



# STIC Search Report

## EIC 2100

STIC Database Tracking Number: 200189

TO: Michael Pham  
Location: RND 3D18  
Art Unit: 2167  
Thursday, August 31, 2006

Case Serial Number: 10/659642

From: Emory Damron  
Location: EIC 2100  
RND 4B19  
Phone: 571-272-3520

Emory.Damron@uspto.gov

### Search Notes

Dear Michael,

Please find below your fast and focused search.

References of potential pertinence have been tagged, but please review all the packets in case you like something I didn't.

Of those references which have been tagged, please note any manual highlighting which I've done within the document.

There may be a few decent references contained herein, but I'll let you determine how useful they may be to you.

Please contact me if I can refocus or expand any aspect of this case, and please take a moment to provide any feedback (on the form provided) so EIC 2100 may better serve your needs. Good Luck!

Sincerely,

Emory Damron

Technical Information Specialist

EIC 2100, US Patent & Trademark Office

Phone: (571) 272-3520

Emory.damron@uspto.gov

915034398778





200189

# STIC EIC 2100 Search Request Form

183

Today's Date: 8/30/2006 What date would you like to use to limit the search?  
Priority Date: 9/10/2002 Other:

Name <u>Michael Pham</u>	Format for Search Results (Circle One): <u>PAPER</u> DISK EMAIL
AU <u>2167</u> Examiner # <u>81563</u>	Where have you searched so far?
Room # <u>3D18</u> Phone <u>23924</u>	USP DWPI EPO JPO ACM IBM TDB
Serial # <u>10/659642</u>	IEEE INSPEC SPI Other <u>EAST</u>

Is this a "Fast & Focused" Search Request? (Circle One) YES NO  
A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria. The criteria are posted in EIC2100 and on the EIC2100 NPL Web Page at <http://ptoweb/patents/stic/stic-tc2100.htm>.

What is the topic, novelty, motivation, utility, or other specific details defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please attach a copy of the abstract, background, brief summary, pertinent claims and any citations of relevant art you have found.

Is this request for a BOARD of APPEALS case? (Circle One) YES NO

Looking for the below limitations:  
receiving at a destination server  
metadata and a set of stub files  
associated with the set of files

looking  
Primarily for maintaining a list of repository nodes that are associated  
with each file in the set of files by updating  
a location components in the file server

App: migrates data in event of a disaster

EXAGRID

Keywords: HSM (Hierarchical storage management), disaster recovery, stub files

STIC Searcher Sharon Darrin Phone 23520  
Date picked up 8/30/6 Date Completed 8/31/6

Xcopy



2004-88382



# STIC Search Results Feedback Form

**EIC 2100**

Questions about the scope or the results of the search? Contact *the EIC searcher or contact:*

Alyson Dill, EIC 2100 Team Leader  
272-3527, RND 4B28

## Voluntary Results Feedback Form

➤ I am an examiner in Workgroup: 2167 Example: 2133

➤ Relevant prior art **found**, search results used as follows:

- ☐ 102 rejection
- ☐ 103 rejection
- ☐ Cited as being of interest.
- ☐ Helped examiner better understand the invention.
- ☐ Helped examiner better understand the state of the art in their technology.

Types of relevant prior art found:

- ☐ Foreign Patent(s)
- ☐ Non-Patent Literature  
(Journal articles, conference proceedings, new product announcements etc.)

➤ Relevant prior art **not found**:

- ☐ Results verified the lack of relevant prior art (helped determine patentability).
- ☐ Results were not useful in determining patentability or understanding the invention.

**Comments:**

Drop off or send completed forms to STIC/EIC2100 RND, 4B28



Set	Items	Description
S1	307975	FILESERVER? OR SERVER OR REPOSITOR? OR DATABANK? OR DATABASE? OR DATASERVER?
S2	2796	DATA() (SERVER? OR ARCHIV?)
S3	1480	DATAFILE? OR DATA() (REPOSITOR? OR HISTOR? OR COMPILATION? - OR ARCHIV?)
S4	188366	DASTOR? OR DATA() (BASE? OR STORAG? OR DEPOSITOR? OR FILE? OR BANK? OR RECORD?)
S5	16293	DATARECORD? OR DATA() (REGIST? OR FIELD?)
S6	1248	DATAREPOSIT? OR DATARECORD? OR DATASET?
S7	3989987	MIGRAT? OR SEND??? OR SENT OR TRANSFER? OR TRANSLOCAT? OR - MOVE? OR MOVING?
S8	3852359	TRANSMIT? OR TRANSMIS? OR SHIP???? OR MAIL???? OR RECEIV?
S9	4903522	ROUTE? OR MAP? ? OR MAPP???? OR REROUT? OR TUNNEL? OR LINK? OR HYPERLINK? OR CONNECT?
S10	131934	REDIRECT? OR FUNNEL? OR SHUNT? OR MIGRAT? OR RELOCAT? OR T-RANSMIGRAT?
S11	626291	DISPATCH? OR UPLOAD? OR DOWNLOAD? OR FORWARD? OR POST???
S12	1460183	IMPORT? OR EXPORT? OR TRANSPORT? OR BACKUP? OR BACK?()UP OR UPDAT? OR REFRESH?
S13	351654	S1:S6 AND S7:S12
S14	5890	STUB?? OR STUBB??? OR ABBREVIAT? OR BIBLIOGRAPHIC? OR SHOR-TEN? OR ZIP OR ZIPS
S15	9863	ZIPPING OR ZIPPING OR META OR COMPRESS? OR CONDENS? OR ABR-IDG? OR SKELETON?
S16	59470	MINIMIZ? OR MINIMIS? OR REDUC? OR CURTAIL? OR TRUNCAT? OR - MINIF? OR DECREAS?
S17	326937	FILE? OR DATA? OR OBJECT? OR INFORMATION?
S18	24170	DATAFILE? OR FOLDER? OR DATAFOLDER? OR DIRECTORYFILE? OR D-ATA()RECORD?
S19	1062	COMPUTERFILE? OR DATARECORD? OR DATA() (TABLE? OR COLUMN? OR ROW? ?)
S20	29187	NAME? OR EXTENSION? OR DESIGNATION? OR LABEL? OR METADATA? OR META()DATA?
S21	37347	IDENTIFIER? OR POINTER? OR INDICATER? OR INDICATOR? OR MAR-KER? OR ID
S22	11168	TITLE? OR IDENTIT? OR BRAND? OR APPELLATION? OR SUFFIX? OR PREFIX?
S23	122964	SOURCE? OR PRIMARY OR ORIGIN? OR LOCAL? OR ONSITE? OR ON()-SITE OR FIRST? OR 1ST OR MAIN OR CHIEF? OR INITIAL?
S24	52514	TARGET? OR DESTINATION? OR REMOTE? OR OFFSITE? OR OFF()SITE
S25	93546	SECOND? OR 2ND OR DIFFERENT? OR ANOTHER? OR ENDPOINT?
S26	21298	LIST? OR CONCORDANC? OR SPREADSHEET?
S27	18631	CHART? ? OR INDEX? OR DIRECTORY? OR ARCHIV? OR LEGEND? ? OR (LOOKUP OR LOOK()UP)()TABLE? OR LUT OR LUTS
S28	3294	(CROSS OR X)()REFERENC? OR CROSSREFERENC? OR XREFERENC? OR CHRONICL? OR ACCOUNTING?
S29	7639	LOCATER? OR CATALOG? OR LIBRAR?
S30	110317	LOCATION? OR ADDRESS? OR SITE? ? OR AREA? OR ZONE? OR LOCA-LE? OR LOCALIT? OR REGION? OR VICINIT?
S31	66951	COMPONENT? OR CONTENT? OR SUBSTANCE?
S32	213483	IC=G06F?
S33	186611	MC=T01?
S34	32023	S13 AND S7:S12(10N)S1:S6 AND (S14:S16(5N)S17:S19 OR S20:S2-2(5N)S17:S19)
S35	2575	S34 AND S23(7N)S1:S6 AND S24:S25(7N)S1:S6
S36	4887	S34 AND S14:S16 AND S20:S22
S37	477	S35 AND S36
S38	435	S37 AND S32:S33
S39	477	S37:S38
S40	158	S39 AND S26:S29



S41 114 S40 AND S30:S31  
 S42 158 S40:S41  
 S43 198 S37 AND S14:S16(5N)S17:S19 AND S20:S22(5N)S17:S19 AND S17:-  
       S19(7N)S7:S12  
 S44 172 S43 AND S23(5N)S1:S6 AND S24:S25(5N)S1:S6 AND (S1:S6 OR S1-  
       7:S19) (5N)S7:S12  
 S45 6 42 AND S43  
 S46 282 S42:S45  
 S47 252 S46 AND S7:S12(5N)S14:S22 AND S7:S12(5N)S1:S6  
 S48 213 S47 AND S23(5N)S1:S6 AND S24:S25(5N)S1:S6  
 S49 4 S46 AND (HIERARCHIC?())(STORAG? OR MANAG?) OR DISASTER? OR -  
       CRASH? OR CALAMIT? OR ACCIDENT? OR MISHAP?)  
 S50 215 S48:S49  
 S51 164 S50 AND AC=US/PR  
 S52 143 S51 AND AY=(1970:2002)/PR  
 S53 111 S51 NOT AY=(2003:2006)/PR  
 S54 51 S50 NOT S51  
 S55 39 S54 AND AY=1970:2002  
 S56 35 S54 NOT AY=2003:2006  
 S57 188 S52:S53 OR S55:S56  
 S58 188 IDPAT (sorted in duplicate/non-duplicate order)  
 File 347:JAPIO Dec 1976-2005/Dec(Updated 060404)  
       (c) 2006 JPO & JAPIO  
 File 350:Derwent WPIX 1963-2006/UD=200655  
       (c) 2006 The Thomson Corporation

57/3,K/40 (Item 34 from file: 350)  
DIALOG(R) File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

✓ ERY  
GOOD

0013366353 - Drawing available  
WPI ACC NO: 2003-455777/200343  
Related WPI Acc No: 2003-541012; 2005-617339  
XRPX Acc No: N2003-362387

**File back up system in network server, stores back up file and meta data representative of location of back up file in order to create index for accessing back up file**

Patent Assignee: LIVEVAULT CORP (LIVE-N)  
Inventor: CAMPBELL J; CHATIER D; GONSALVES M; MIDGLEY C; WEBB J  
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 6526418	B1	20030225	US 1999465411	A	19991216	200343 B

Priority Applications (no., kind, date): US 1999465411 A 19991216

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6526418	B1	EN	21	7	

**...system in network server, stores back up file and meta data representative of location of back up file in order to create index for accessing back up file**

#### Original Titles:

Systems and methods for backing up data files

**Alerting Abstract ...NOVELTY** - A dynamic replication process in back - up server , records changes in a journal file made by an operating agent on the data server . One version of the back up file is then stored in a long term recording medium like tape. A catalog process records meta data representative of the locations of one version of the back up file on the storage medium, to create an index for accessing the back up file . USE - For maintaining back up files for server on computer network...

**...ADVANTAGE** - Provides a safe point signal for determining when to store an amended target data file to a long term medium such as tape. Allows user to choose network consumption limit during working hours and allows unlimited consumption at other times. Provides continuous back up of data stored on the computer network. Provides a secure restore process for determining access right of a party that is requesting to restore a version of a target data file .

...

**...DESCRIPTION OF DRAWINGS** - The figure shows the flowchart explaining the process for synchronizing identified source data files with target data files to create a baseline data structure at the backup server

Title Terms.../Index Terms/Additional Words: META ; ...

... INDEX ;

#### Class Codes

International Classification (Main): G06F-017/30

Manual Codes (EPI/S-X): T01-F05E ...

... T01-G03 ...

... T01-N02B1A

#### Original Publication Data by Authority

##### Original Abstracts:

The invention provides systems and methods for continuous **back up** of **data** stored on a computer network. To this end the systems of the invention include a synchronization process that replicates selected **source data files** data stored on the network and to create a corresponding set of replicated **data files**, called the target **data files**, that are stored on a **back up server**. This synchronization process builds a baseline **data** structure of **target data files**. IN parallel to this synchronization process, the system includes a dynamic replication process that includes a plurality of agents, each of which monitors a portion of the **source data files** to detect and capture, at the byte-level, changes to the **source data files**. Each agent may record the changes to a respective journal file, and as the dynamic replication process detects that the journal **files** contain data, the journal files are **transferred** or copied to the **back up server** so that the captured changes can be written to the appropriate ones of the **target data files**.

##### Claims:

We claim:1. A system for **backing up** at least one **source data file**, comprising a **data server** having storage for the at least **one source data** file, and an agent operating on the data server and capable of dynamically detecting changes to the at least **one source data** file and capable of recording detected changes within a **journal file**, a **back up server** having storage for at least **one back up** file, a dynamic replication process, **responsive** to data within the journal file for writing the changes to the at least **one back up** file to mirror changes made to the at least **one source data** file, a long term storage system for recording that at least **one back up** file to a storage medium in response to the operation of the dynamic replication process, thereby storing at least one version of the at least **one back up** file on the storage **medium**, and a catalog **process** for recording **metadata** representative of locations of the at least one version of the at least **one back up** file on the storage medium, to create an index for accessing the at least one version of the at least **one back up** file.



US006526418B1

(12) **United States Patent**  
**Midgley et al.**

(10) **Patent No.: US 6,526,418 B1**  
(45) **Date of Patent: Feb. 25, 2003**

(54) **SYSTEMS AND METHODS FOR BACKING UP DATA FILES**

(75) **Inventors:** Christopher Midgley, Framingham, MA (US); John Webb, Sutton, MA (US); Jim Campbell, Littleton, MA (US); Manuel Gonsalves, Milford, MA (US); Dan Chatier, Woonsocket, RI (US)

(73) **Assignee:** LiveVault Corporation, Marlborough, MA (US)

(\*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) **Appl. No.:** 09/465,411

(22) **Filed:** Dec. 16, 1999

(51) **Int. Cl.<sup>7</sup>** ..... G06F 17/30; H04B 1/74

(52) **U.S. Cl.** ..... 707/204; 707/8; 707/202; 707/203; 709/236; 711/162; 713/176; 714/4; 714/5

(58) **Field of Search** ..... 707/1, 10, 100, 707/202, 204, 8, 203; 711/162; 714/5, 2, 4; 713/201, 176; 709/214-216, 236

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

4,751,635 A	6/1988	Kret	364/200
5,043,871 A	8/1991	Nishigaki et al.	364/200
5,572,709 A	11/1996	Fowler et al.	395/500
5,574,898 A	11/1996	Leblang et al.	707/1
5,649,200 A	7/1997	Leblang et al.	395/703
5,675,802 A	10/1997	Allen et al.	395/703
5,745,753 A	4/1998	Mosher, Jr.	395/618
5,754,782 A	5/1998	Masada	395/200.43
5,758,359 A *	5/1998	Saxon	707/204
5,764,972 A	6/1998	Crouse et al.	707/1
5,778,395 A	7/1998	Whiting et al.	707/204
5,794,563 A	8/1998	Klepac	119/75.91

5,854,754 A	12/1998	Cabrera et al.	364/578
5,920,701 A	7/1999	Miller et al.	395/200.58
5,931,947 A	8/1999	Burns et al.	713/201
5,974,563 A *	10/1999	Beeler, Jr.	707/204
5,991,772 A	11/1999	Doherty et al.	707/202

(List continued on next page.)

**FOREIGN PATENT DOCUMENTS**

WO WO 98/54662 12/1998 ..... G06F/17/30

**OTHER PUBLICATIONS**

Wolfsom, Ouri et al., "Distributed Algorithms For Dynamic Replication of Data", Proceedings of the 11<sup>th</sup> ACM-SIGMOD—symposium on Principles of Database Systems, Jun. 1992, pp. 149–163.

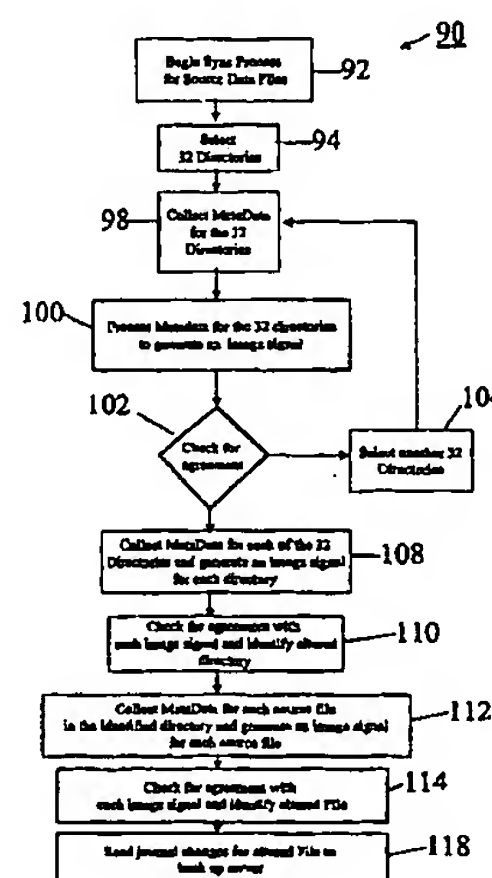
*Primary Examiner*—Shahid Alam

(74) *Attorney, Agent, or Firm*—Kevin A. Oliver; Foley Hoag LLP

(57) **ABSTRACT**

The invention provides systems and methods for continuous back up of data stored on a computer network. To this end the systems of the invention include a synchronization process that replicates selected source data files data stored on the network and to create a corresponding set of replicated data files, called the target data files, that are stored on a back up server. This synchronization process builds a baseline data structure of target data files. IN parallel to this synchronization process, the system includes a dynamic replication process that includes a plurality of agents, each of which monitors a portion of the source data files to detect and capture, at the byte-level, changes to the source data files. Each agent may record the changes to a respective journal file, and as the dynamic replication process detects that the journal files contain data, the journal files are transferred or copied to the back up server so that the captured changes can be written to the appropriate ones of the target data files.

**11 Claims, 6 Drawing Sheets**





23

in C., Hayden Publishing (1983) Moreover, although the figures graphically depict the agent processes 30, synchronization replication process 40 and other processes as functional block elements, it will be apparent to one of ordinary skill in the art that these elements can be realized as computer programs or portions or computer programs that are capable of running on the data processor platforms to thereby configure the data processors as a system according to the invention.

Those skilled in the art will know or be able to ascertain using no more than routine experimentation, many equivalents to the embodiments and practices described herein. Accordingly, it will be understood that the invention is not to be limited to the embodiments disclosed herein, but is to be understood from the following claims, which are to be interpreted as broadly as allowed under the law.

We claim:

1. A system for backing up at least one source data file, comprising a data server having
  - storage for the at least one source data file, and
  - an agent operating on the data server and capable of dynamically detecting changes to the at least one source data file and capable of recording detected changes within a journal file,
  - a back up server having storage for at least one back up file,
  - a dynamic replication process, responsive to data within the journal file for writing the changes to the at least one back up file to mirror changes made to the at least one source data file,
  - a long term storage system for recording that at least one back up file to a storage medium in response to the operation of the dynamic replication process, thereby storing at least one version of the at least one back up file on the storage medium, and
  - a catalog process for recording metadata representative of locations of the at least one version of the at least one back up file on the storage medium, to create an index for accessing the at least one version of the at least one back up file.
2. A system according to claim 1, wherein the catalog process includes means for storing security metadata associated with the at least one version of the at least one back up file and representative of user access rights for the at least one version of the at least one back up file.
3. A system according to claim 2, further including a secure restore process for determining the access rights of a party requesting to restore a version of the at least one back up file, and for comparing the access right of the party with the access rights associated with the version of the at least one back up file to be restored.
4. A system according to claim 3, wherein the secure restore process includes means, responsive to the comparison of access rights of the party and access rights associated with the version of the at least one back up file, to determine whether the party may view metadata associated with the version of the at least one back up file.
5. A system according to claim 3, wherein the secure restore process includes means, responsive to the comparison of access rights of the party and access rights associated with the version of the at least one back up file, to determine whether the party may restore the version of the at least one back up file.

24

6. A system for creating back up files for data files stored on a computer network, comprising
  - a data server having a source data file,
  - a backup process for identifying changes in the source data file and for writing the changes to a target data file to mirror changes made to the source data file,
  - a long term storage system for recording the target file to a storage medium in response to the operation of the back up process, thereby storing versions of the target file on the storage medium,
  - a catalog process for recording metadata representative of locations of the versions of the target files on the storage medium and security data associated with the versions of the target data file and representative of user access rights for the versions of the target data file, and
  - a secure restoration process for employing the security data to analyze access rights in response to a request to access the versions of the target file.
7. A system for backing up at least one source data file, comprising
  - a data server having storage for at least one source data file,
  - a back up process for identifying changes in the at least one source data file and for writing the changes to at least one back up file to mirror changes made to the at least one source data file,
  - a long term storage system for recording the at least one back up file to a storage medium in response to the operation of the back up process, thereby storing at least one version of the at least one back up file on the storage medium,
  - a catalog process for recording metadata representative of locations of the at least one version of the at least one back up file on the storage medium and security data associated with the at least one version of the at least one back up file and representative of user access rights for the at least one version of the at least one back up file, and
  - a secure restoration process for employing the security data to analyze access right in response to a request to access the at least one version of the at least one back up file.
8. A system for backing up at least one source data file, the system comprising:
  - a data serving having
    - storage for the at least one source data file, and
    - an agent operating on the data server and capable of dynamically detecting changes to the at least one source data file and capable of recording detected changes within a journal file;
  - a back up server having storage for at least one back up file;
  - a dynamic replication process responsive to data within the journal file for modifying the at least one back up file based on the journal file;
  - a long term storage system for recording the at least one back up file to a storage medium in response to the operation of the dynamic replication process, thereby storing at least one version of the at least one back up file on the storage medium; and
  - a catalog process for recording metadata representative of locations of the at least one version of the at least one back up file on the storage medium, to create an index for accessing the at least one version of the at least one

25

back up file, wherein the catalog process is capable of storing securing metadata associated with the at least one version of the at least one back up file and representative of user access rights for the at least one version of the at least one back up file.

9. The system of claim 8, further comprising:

a secure restore process for determining the access rights of a party requesting to restore a version of the at least one back up file and comparing the access rights of the party with the access rights associated with the version of the at least one back up file to be restored.

10. The system of claim 9, wherein responsive to the comparison of access rights of the party and access rights

26

associated with the version of the at least one back up file, the secure restore process is capable of determining whether the party may view metadata associated with the version of the at least one back up file.

5 11. The system of claim 9, wherein, responsive to the comparison of access rights of the party and access rights associated with the version of the at least one back up file, the secure restore process is capable of determining whether  
10 the party may restore the version of the at least one back up file.

\* \* \* \* \*

57/3,K/50 (Item 44 from file: 350)  
DIALOG(R) File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013182750

WPI ACC NO: 2003-266196/200326

Related WPI Acc No: 2002-566953; 2003-074756; 2003-090905; 2003-102825;  
2003-102942; 2003-110871; 2003-156363; 2003-174622; 2003-174624;  
2003-198567; 2003-198602; 2003-198603; 2003-198604; 2003-199853;  
2003-209442; 2003-238461; 2003-265798; 2003-276087; 2003-276243;  
2003-288711; 2005-505116; 2006-171305; 2006-442695

XPX Acc No: N2003-211403

**Distributed file storage system for personal computer network, has pair of network connected file servers, into which file system metadata are loaded to allow client computer to locate store files**

Patent Assignee: BOLSTAD G D (BOLS-I); PRIESTER W G (PRIE-I); RANDALL J G (RAND-I); SCHWEITZER J R (SCHW-I); STAUB J R (STAU-I); ULRICH T R (ULRI-I)

Inventor: BOLSTAD G D; PRIESTER W G; RANDALL J G; SCHWEITZER J R; STAUB J R; ULRICH T R

**Patent Family** (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20020178162	A1	20021128	US 2001264668	P	20010129	200326 B
			US 2001264669	P	20010129	
			US 2001264670	P	20010129	
			US 2001264671	P	20010129	
			US 2001264672	P	20010129	
			US 2001264673	P	20010129	
			US 2001264694	P	20010129	
			US 2001302424	P	20010629	
			US 200260858	A	20020129	

Priority Applications (no., kind, date): US 2001302424 P 20010629; US 2001264694 P 20010129; US 2001264673 P 20010129; US 2001264672 P 20010129; US 2001264671 P 20010129; US 2001264670 P 20010129; US 2001264669 P 20010129; US 2001264668 P 20010129; US 200260858 A 20020129

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20020178162	A1	EN	96	44	Related to Provisional US 2001264668
					Related to Provisional US 2001264669
					Related to Provisional US 2001264670
					Related to Provisional US 2001264671
					Related to Provisional US 2001264672
					Related to Provisional US 2001264673
					Related to Provisional US 2001264694
					Related to Provisional US 2001302424

**Distributed file storage system for personal computer network, has pair of network connected file servers, into which file system metadata are loaded to allow client computer to locate store files**

**Alerting Abstract ...NOVELTY** - The system has a pair of file servers which are operably connected to the network. The portions of file system metadata which organizes file data into parity groups, are

loaded into corresponding **file** servers, where **metadata** specifies the **directory** structure that aggregates files stored by corresponding servers, to allow client computer to locate stored...

...a **data storage** method; and a computer network file system...

...USE - In high-performance mass storage systems of personal computer network, for **data storage**, **backup**, recovery, where the system uses redundant array of independent disks (RAID) technology...

...the user can freely access any file without any specific knowledge on the current physical **location** of the **file**. The system **reduces** the maintenance and monitoring requirements. The architecture provides distributed parity groups that are integrated into distributed file storage system technology, to optimize use of disk resources by **moving** frequently and infrequently accessed **data** blocks between drives to maximize throughput and capacity utilization of each drive. Also increases storage ...

**Title Terms...**/Index Terms/Additional Words: **CONNECT** ;

#### **Class Codes**

International Classification (+ Attributes)

IPC + Level Value Position Status Version

G06F-0015/16 ...

... G06F-0007/00

G06F-0015/16 ...

... G06F-0007/00

Manual Codes (EPI/S-X): **T01-H01B1A** ...

... **T01-J05B2** ...

... **T01-N02A2C** ...

... **T01-N03A2**

#### **Original Publication Data by Authority**

#### **Original Abstracts:**

...file system with variable parity groups is described. The integrated distributed file system includes a **first file server** that is operably connected to a network fabric and a second **file server** that is operably **connected** to the network fabric. The integrated distributed **file system** further includes **file system metadata** that includes a first portion of the **file system metadata** that is loaded on the **first file server** and a **second** portion of the **file system metadata** that is loaded on the **second file server**. The **file system metadata** specifies at least a portion of a **directory** structure that aggregates files stored by the **first file server** and by the **second file server** to allow a client computer that is operably **connected** to the network fabric to locate **files** stored by the **first file server** and files stored by the **second file server** without prior knowledge as to which **file server** stores a desired **file**. The **file system metadata** further organizes **file data** into distributed parity groups. Each distributed parity group includes one or more data blocks and a parity block. The **file system metadata** includes disk **identifier data** and disk address **data** for



each of the data blocks and for the parity block.

**Claims:**

What is claimed is: b 1 /b . A distributed file storage system, comprising: a **first** file server operably connected to a network fabric; a second file server operably connected to said network fabric; **file system** metadata, a first portion of said **file system metadata** loaded on said **first file server** ; and a second portion of said **file system** metadata loaded on said **second file server**, said **file system** metadata specifying at least a portion of a directory structure that aggregates files stored by said **first file server** and said **second file server** to allow a client computer **operably** connected to said network fabric to **locate** files stored by said **first file server** and files stored by said **second file server** without prior knowledge as to which **file server** stores a **desired** file, said **file system** metadata further **organizing** **file** data into distributed parity groups, each distributed parity group comprising one or more data blocks and a parity block, said **file system** metadata comprising **disk identifier** data and **disk address** data for each of said data blocks and said parity block.

57/3,K/39 (Item 33 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013448729 - Drawing available  
WPI ACC NO: 2003-539984/200351  
XRPX Acc No: N2003-428226

Database change propagation method in database management system,  
involves migrating data from source table to target table by executing  
multiple select statements

Patent Assignee: CORIGIN LTD (CORI-N)

Inventor: ROTHSCCHILD M

Patent Family (1 patents, 1 countries)

Patent

Application

Number	Kind	Date	Number	Kind	Date	Update
US 6567823	B1	20030520	US 2000633332	A	20000807	200351 B

Priority Applications (no., kind, date): US 2000633332 A 20000807

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6567823	B1	EN	21	13	

Database change propagation method in database management system,  
involves migrating data from source table to target table by executing  
multiple select statements

**Alerting Abstract ...NOVELTY** - An independent key representing migration  
of source table is obtained from log file (1). A condensed table (4)  
is generated with records representing each data row changed in source  
table according to key. A file containing insert, update and delete  
statements (8-10) is generated for each source table. A target table (15)  
is updated based on commands in insert statement file generated from  
respective source table. USE - For propagating change during migration of  
source table to target table in database management system (DBMS...)

...ADVANTAGE - Enables propagation change by migrating source table to  
target table, thereby avoiding rescanning and recreating source and target  
table. Performs...

...DESCRIPTION OF DRAWINGS - The figure shows an explanatory view of the  
database change propagation method...

...4 condensed table...

...9 update statement...

Title Terms/Index Terms/Additional Words: DATABASE ; ...

... MIGRATION ;

Original Publication Data by Authority

#### Original Abstracts:

A change propagation method for a migration from a source table to a  
target table in a DBMS which uses the log...

7  
AUG  
2000  
Filing

...this log file a last data-independent key is derived to represent an immediate previous **migration** and a **condense** table is generated with records representing before and after images of a data row which has changed in the source table. For each source table in a **migration** definition **files** containing delete statements, **update** statements, an insert table and an ignore **file** are provided. A target table is **updated** based upon commands in an insert statement **file** generated from the **migration** statement, the ignore **file**, the source table and the insert table.

**Claims:**

I claim:1. A method of change propagation in a **migration** from at least one **source** table to a **target** table in a **database** management system for a **database** containing said **source** table and having a log in which each specific record has a data-independent key...

...records than for older records, the method comprising:for supporting a change-propagation process for **migrations** that use more than one source-table instance for single-table subcondition traceable **migrations**,for supporting a change-propagation process for fixed environment traceable **migrations**,for supporting a change-propagation process for fixed environment traceable **migrations** without relying on a stable version of an optionally changeable source table, orfor supporting a change-propagation process for single-table **instance** migrations without relying on a stable version of an optionally changeable source table,the steps of:(a) deriving from said log of **said** database containing **said** source table a **last** data-independent key representing an immediately **previous** migration;(b) for a **particular** migration definition and the respective **last** data-independent key derived in step (a), for a source table undergoing a change involving a migration, generating a condensed table containing at most two records for a **record** identifier of **each** **data** row which has changed in the source table after the completion of a **previous** migration or change propagation process, the two records being selected from before and after images...

...pair of images are formed with each change;(c) for each source table appearing in **said** migration definition, eliminating redundant changes and from **said** migration definition and **the** condensed table, generating a file containing delete statements, a file **containing** update statements, an insert table and an **ignore** file containing instructions to ignore specific rows in a table;(d) thereafter:(d1) implementing deletes based upon **the** file containing delete statements **and** updating the target table correspondingly,(d2) **implementing** updates based upon **the** file **containing** update statements **and** updating the target table correspondingly,(d3) where a single source table is involved in **said** migration definition and **said** migration definition does not involve an SQL join step, implementing insert commands contained in said insert table **and** updating the target table correspondingly, and(d4) where a plurality of source tables are involved in **said** migration definition and **said** migration definition includes an SQL join step, joining each insert table generated from one of...

...least parts of the other source tables of said plurality of source tables based upon **the** migration definition, the **ignore** file, the respective source table and the insert table to generate a respective insert **statement** file; and( e ) updating said target table based upon commands in said insert statement file.

57/3,K/37 (Item 31 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013504847 - Drawing available

WPI ACC NO: 2003-597412/

XRPX Acc No: N2003-476138

**Tailored classified file deposit screen for transferring specific  
classified file between computers in workflow system, transfers tailored  
classified file and classification identifier to file server of  
computer network**

Patent Assignee: SORIANO E M (SORI-I); YUAN T C (YUAN-I)

Inventor: SORIANO E M; YUAN T C

**Patent Family** (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
US 20030097399	A1	20030522	US 2001989776	A	20011119	200356	B

Priority Applications (no., kind, date): US 2001989776 A 20011119

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20030097399	A1	EN	24	14	

**Tailored classified file deposit screen for transferring specific  
classified file between computers in workflow system, transfers tailored  
classified file and classification identifier to file server of  
computer network**

#### Original Titles:

Workflow system for transferring a specific classified file

**Alerting Abstract** ...NOVELTY - A web page displays the specification of a tailored classified file. A classification identifier identifies the file on storage device connected to a computer network. The tailored classified file and the classification identifier are transferred to a file server of another computer, after identification....USE - For transferring specific classified files in a workflow system in off-line business activities such as travel expense approval, purchasing of catalog items and product design process...

...and classified nature of the screens permits easy understanding and use by the users. Hence, minimizes the training required by the users...

...DESCRIPTION OF DRAWINGS - The figure shows the detailed block diagram of the specific classified file transfer workflow system.

**Title Terms**.../Index Terms/Additional Words: **TRANSFER** ;

#### Class Codes

International Classification (Main): G06F-015/16

Manual Codes (EPI/S-X): T01-F05E ...

... T01-S03

Original Publication Data by Authority

19  
NOV  
2001  
Filing



#### Original Abstracts:

A tailored classified file deposit screen is provided by a **server connected** to a **first** computer network. The tailored classified **file** deposit screen is configured to **transfer** a specific classified **file** based on a step in a sequence of steps of a process. A display of a specification of the file is provided. A means to identify the **file** on a storage device **connected** to a second computer network is also provided. Finally a means to **transfer** the **file** and a classification **identifier** to the **server** is provided.

#### Claims:

What is claimed is:     b 1 /b . A tailored classified file deposit screen provided by a **server connected** to a **first** computer network wherein the tailored classified **file** deposit screen is configured to **transfer** a specific classified **file** , based on a step in a sequence of steps of a process, providing the following: a display of a specification of the file; means to identify the **file** on a storage device **connected** to a second computer network; and means to **transfer** the **file** and a classification **identifier** to the **server**.

57/3,K/34 (Item 28 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013577000 - Drawing available  
WPI ACC NO: 2003-671567/200363  
XRPX Acc No: N2003-536179

Metadata retrieval method for multimedia objects , utilizes a database  
of fingerprints generated from multimedia objects transmitted on a  
transmission channel

Patent Assignee: HAITSMA J A (HAIT-I); KONINK PHILIPS ELECTRONICS NV  
(PHIG)

Inventor: HAITSMA A; HAITSMA J A

Patent Family (11 patents, 101 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
WO 2003067467	A1	20030814	WO 2003IB260	A	20030127	200363 B
AU 2003202728	A1	20030902	AU 2003202728	A	20030127	200425 E
EP 1474760	A1	20041110	EP 2003701639	A	20030127	200473 E
			WO 2003IB260	A	20030127	
KR 2004081191	A	20040920	KR 2004712243	A	20040806	200508 E
JP 2005517245	W	20050609	JP 2003566746	A	20030127	200538 E
			WO 2003IB260	A	20030127	
US 20050144455	A1	20050630	WO 2003IB260	A	20030127	200543 E
			US 2004503685	A	20040804	
CN 1628303	A	20050615	CN 2003803302	A	20030127	200563 E
EP 1474760	B1	20051207	EP 2003701639	A	20030127	200582 E
			WO 2003IB260	A	20030127	
DE 60302651	E	20060112	DE 60302651	A	20030127	200613 E
			EP 2003701639	A	20030127	
			WO 2003IB260	A	20030127	
IN 200401699	P4	20060224	US 2002364322	P	20020314	200619 E
			IN 2004CN1699	A	20040803	
DE 60302651	T2	20060810	DE 60302651	A	20030127	200654 E
			EP 2003701639	A	20030127	
			WO 2003IB260	A	20030127	

Priority Applications (no., kind, date): EP 200275501 A 20020206

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
--------	------	-----	----	-----	--------------

WO 2003067467	A1	EN	24	3	
---------------	----	----	----	---	--

National Designated States,Original: AE AG AL AM AT AU AZ BA BB BG BR BY  
BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID  
IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ  
NO NZ OM PH PL PT RO RU SC SD SE SG SK SL TJ TM TN TR TT TZ UA UG US UZ  
VC VN YU ZA ZM ZW

Regional Designated States,Original: AT BE BG CH CY CZ DE DK EA EE ES FI  
FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PT SD SE SI SK SL SZ TR  
TZ UG ZM ZW

AU 2003202728	A1	EN			Based on OPI patent WO 2003067467
EP 1474760	A1	EN			PCT Application WO 2003IB260

Based on OPI patent WO 2003067467

Regional Designated States,Original: AL AT BE BG CH CY CZ DE DK EE ES FI  
FR GB GR HU IE IT LI LT LU LV MC MK NL PT RO SE SI SK TR

JP 2005517245	W	JA	18		PCT Application WO 2003IB260
---------------	---	----	----	--	------------------------------

Based on OPI patent WO 2003067467

US 20050144455	A1	EN	PCT Application	WO 2003IB260
EP 1474760	B1	EN	PCT Application	WO 2003IB260
			Based on OPI patent	WO 2003067467
Regional Designated States, Original: AT BE BG CH CY CZ DE DK EE ES FI FR				
GB GR HU IE IT LI LU MC NL PT SE SI SK TR				
DE 60302651	E	DE	Application	EP 2003701639
			PCT Application	WO 2003IB260
			Based on OPI patent	EP 1474760
			Based on OPI patent	WO 2003067467
IN 200401699	P4	EN	PCT Application	US 2002364322
DE 60302651	T2	DE	Application	EP 2003701639
			PCT Application	WO 2003IB260
			Based on OPI patent	EP 1474760
			Based on OPI patent	WO 2003067467

Metadata retrieval method for multimedia objects , utilizes a database of fingerprints generated from multimedia objects transmitted on a transmission channel

#### Original Titles:

...FAST HASH-BASED MULTIMEDIA OBJECT METADATA RETRIEVAL...

...FAST HASH-BASED MULTIMEDIA OBJECT METADATA RETRIEVAL...

...Fast hash-based multimedia object metadata retrieval...

...FAST HASH-BASED MULTIMEDIA OBJECT METADATA RETRIEVAL...

**Alerting Abstract** ...NOVELTY - A metadata retrieval system for multimedia objects utilizes a secondary database (122) storing fingerprints generated by a server (120) from transmissions received on a transmission channel. The server uses reference fingerprints, for multimedia and associated metadata , stored in a primary database (121) to generate the entries in the secondary database . In response to client requests the server searches the secondary database before the primary database . DESCRIPTION - An INDEPENDENT CLAIM is also included for a system for obtaining metadata for a multimedia object .

...

...USE - For obtaining metadata for multimedia objects by matching an object fingerprint...

...ADVANTAGE - The secondary database contains a small number of entries, all for objects transmitted over a monitored channel, thus matching a client request against the secondary database will be much faster than matching against the primary database , only if no match is found in the secondary database will the primary database be searched. Because requests are more likely to be received for objects transmitted over the monitored transmission channel, the average time required to match a fingerprint is significantly reduced .

...

...DESCRIPTION OF DRAWINGS - The figure schematically shows an embodiment of a server for multimedia metadata retrieval...

...120 Server

...

...121 Primary database

...

...122 Secondary database.

Title Terms.../Index Terms/Additional Words: DATABASE ; ...

... TRANSMIT ; ...

... TRANSMISSION ;

#### Class Codes

International Classification (Main): G06F-012/00 ...

... G06F-017/30 ...

... G06F-017/40

International Classification (+ Attributes)

IPC + Level Value Position Status Version

G06F-0017/30 ...

... G06F-0017/30 ...

... G06F-0017/30

G06F-0017/30 ...

Manual Codes (EPI/S-X): T01-J05B3 ...

... T01-J05B4P ...

... T01-N01D1

#### Original Publication Data by Authority

#### Original Abstracts:

When a client requests **metadata** for a multimedia **object**, a **server** (120) computes a fingerprint for the multimedia object and matches it against entries stored in a **primary database** (121). Additionally, the **server** system (120) monitors one or more **transmission** channels such as radio broadcast channels, and computes **transmission** fingerprints for **transmissions** on those channels. The **transmission** fingerprints are matched with the fingerprints stored in the **primary database** (121), and if a match is found, an entry for the identified **transmission** is then added to the **secondary database** (122). Preferably this entry contains the **transmission** fingerprint. Subsequent matches are then **first** performed against the **secondary database** (122) and only upon unsuccessful matches also matched against the **primary database** (121). Also covers the method and a computer program product for performing the method...

...When a client requests **metadata** for a multimedia **object**, a **server** ( b 120 /b ) computes a fingerprint for the multimedia object and matches it against entries stored in a **primary database** ( b 121 /b ). Additionally, the **server** system ( b 120 /b ) monitors one or more **transmission** channels such as radio broadcast channels, and computes **transmission** fingerprints for **transmissions** on those channels. The **transmission** fingerprints are matched with the fingerprints stored in the



**primary database** ( b 121 /b ), and if a match is found, an entry for the identified **transmission** is then added to the **secondary database** ( b 122 /b ). Preferably this entry contains the **transmission** fingerprint. Subsequent matches are then **first** performed against the **secondary database** ( b 122 /b ) and only upon unsuccessful matches also matched against the **primary database** ( b 121 /b ). Also covers the method and a computer program product for performing the method...

...When a client requests **metadata** for a multimedia **object**, a **server** (120) computes a fingerprint for the multimedia object and matches it against entries stored in a **primary database** (121). Additionally, the **server** system (120) monitors one or more **transmission** channels such as radio broadcast channels, and computes **transmission** fingerprints for **transmissions** on those channels. The **transmission** fingerprints are matched with the fingerprints stored in the **primary database** (121), and if a match is found, an entry for the identified **transmission** is then added to the **secondary database** (122). Preferably this entry contains the **transmission** fingerprint. Subsequent matches are then **first** performed against the **secondary database** (122) and only upon unsuccessful matches also matched against the **primary database** (121). Also covers the method and a computer program product for performing the method...

...donnees primaire (121). De plus, le systeme serveur (120) controle un ou plusieurs canaux de **transmission** tels que des canaux de radiodiffusion, et calcule des empreintes digitales de **transmission** pour les **transmissions** sur lesdits canaux. Les empreintes digitales de **transmission** sont comparees aux empreintes digitales enregistrees dans la base de donnees primaire (121), et si une correspondance est trouvee, une entree pour la **transmission** identifiee est ajoutee a une base de donnees secondaire (122). Ladite entree contient de preference l'empreinte digitale de **transmission**. Des comparaisons suivantes sont ensuite effectuees, tout d'abord avec la base de donnees secondaire...

#### **Claims:**

...A computer-implemented method of obtaining **metadata** for a multimedia **object** in response to a request **received** from a client (110) over a network (115) by matching an object fingerprint for the multimedia object against entries stored in a **primary database** (121) comprising fingerprints and respective associated sets of **metadata**, b characterized by /b computing a **transmission** fingerprint for a portion of a further multimedia **object transmitted** on a monitored **transmission** channel, matching the **transmission** fingerprint with the fingerprints stored in the **primary database** (121), adding an entry comprising at least a fingerprint for the further multimedia object in a **secondary database** (122) upon a successful match, and obtaining the **metadata** by matching the **object** fingerprint against entries stored in the **secondary database** (122) and only matching against the **primary database** (121) if said matching against the **secondary database** (122) fails...

...a une demande recue d'un client (110) sur un reseau (115) en recherchant une **concordance** entre une empreinte d'objet pour l'objet multimedia et des entrees enregistrees dans une...

...et des series associees respectives de metadonnees, b caracterise par /b le calcul d'une empreinte de **transmission** pour une partie d'un objet multimedia complet **transmis** sur un canal de **transmission** controle, la recherche de **concordance** entre l'empreinte de **transmission**

et les empreintes enregistrees dans la base de donnees primaire (121),  
l'addition d'une...

...pour l'objet multimedia complet dans une base de donnees secondaire  
(122) en cas de **concordance** reussie et l'obtention des metadonnees en  
recherchant une **concordance** entre l'empreinte de l'objet et les entrees  
enregistrees dans la base de donnees secondaire (122) et en recherchant une  
**concordance** uniquement dans la base de donnees primaire (121) si ladite  
recherche de **concordance** dans la base de donnees secondaire (122) echoue  
...

...1 /b . A method of obtaining **metadata** for a multimedia **object** by  
matching an object fingerprint for the multimedia object against entries  
stored in a **primary database** , characterized by computing a  
**transmission** fingerprint for a portion of a further multimedia **object**  
**transmitted** on a **transmission** channel, matching the **transmission**  
fingerprint with the fingerprints stored in the **primary database** ,  
adding an entry for the further multimedia object in a **secondary**  
**database** upon a successful match, and obtaining the **metadata** by matching  
the fingerprint against entries stored in the **secondary database** and  
only matching against the **primary database** if said matching against the  
**secondary database** fails.



US 20050144455A1

(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2005/0144455 A1**  
Haitsma (43) **Pub. Date: Jun. 30, 2005**(54) **FAST HASH-BASED MULTIMEDIA OBJECT  
METADATA RETRIEVAL**(52) **U.S. Cl. .... 713/176**(76) **Inventor: Jaap Andre Haitsma, Eindhoven (NL)**(57) **ABSTRACT**

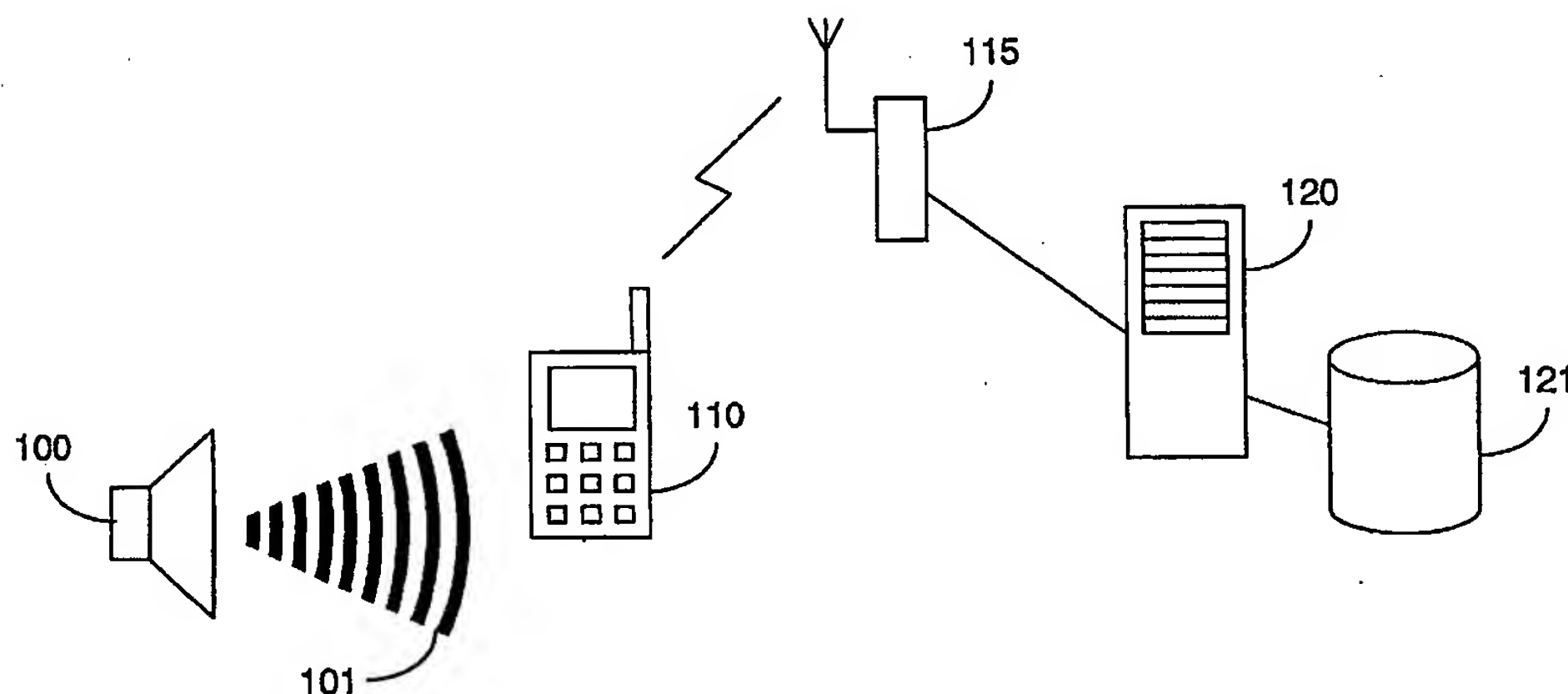
Correspondence Address:

**PHILIPS INTELLECTUAL PROPERTY &  
STANDARDS****P.O. BOX 3001****BRIARCLIFF MANOR, NY 10510 (US)**(21) **Appl. No.: 10/503,685**(22) **PCT Filed: Jan. 27, 2003**(86) **PCT No.: PCT/IB03/00260**(30) **Foreign Application Priority Data**

Feb. 6, 2002 (EP) ..... 02075501.3

**Publication Classification**(51) **Int. Cl.<sup>7</sup> ..... H04L 9/00**

When a client requests metadata for a multimedia object, a server (120) computes a fingerprint for the multimedia object and matches it against entries stored in a primary database (121). Additionally, the server system (120) monitors one or more transmission channels such as radio broadcast channels, and computes transmission fingerprints for transmissions on those channels. The transmission fingerprints are matched with the fingerprints stored in the primary database (121), and if a match is found, an entry for the identified transmission is then added to the secondary database (122). Preferably this entry contains the transmission fingerprint. Subsequent matches are then first performed against the secondary database (122) and only upon unsuccessful matches also matched against the primary database (121). Also covers the method and a computer program product for performing the method.



that those skilled in the art will be able to design many alternative embodiments without departing from the scope of the appended claims.

[0077] For instance, the fingerprint for the multimedia object 101 can alternatively be computed by a fingerprinting module in the mobile phone 110, rather than by the fingerprinting module 202 in the server 120. This way, only the fingerprint itself has to be transmitted to the server 120, and the fingerprinting module 202 can be omitted from the server 120. As the fingerprint is usually smaller than the portion of the multimedia object 101 from which it was computed, this achieves a substantial bandwidth reduction. The construction and operation of a mobile phone equipped with a fingerprinting module is explained in international patent application WO 02/17135 (attorney docket PHNL000469).

[0078] The database 121 could be distributed over multiple physical computers systems, to reduce the workload of each individual system. The contents of the database 121 could also be distributed over a plurality of clients in a file sharing network, as is explained in international patent application PCT/IB02/04605 (attorney docket PHNL010874).

[0079] The contents of the database 121 can be made available for free, or only to paying subscribers. Alternatively, a fee could be charged for every query performed on the database 121. The amount of metadata returned to the client in response to submitting a fingerprint could also be varied: the free service returns only artist and title, and the subscription-based service returns all the metadata available in the database, for example.

[0080] The secondary database 122 can in practice be realized as one or more tables in the primary database 121, although this may make access times to the database slower as the size of the database (preferably held in working memory) now increases.

[0081] In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. The word "comprising" does not exclude the presence of elements or steps other than those listed in a claim. The word "a" or "an" preceding an element does not exclude the presence of a plurality of such elements.

[0082] The invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In the device claim enumerating several means, several of these means can be embodied by one and the same item of hardware. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage.

1. A method of obtaining metadata for a multimedia object by matching an object fingerprint for the multimedia object

against entries stored in a primary database, characterized by computing a transmission fingerprint for a portion of a further multimedia object transmitted on a transmission channel, matching the transmission fingerprint with the fingerprints stored in the primary database, adding an entry for the further multimedia object in a secondary database upon a successful match, and obtaining the metadata by matching the fingerprint against entries stored in the secondary database and only matching against the primary database if said matching against the secondary database fails.

2. The method of claim 1, further comprising receiving at least a portion of the multimedia object and computing the object fingerprint over the received portion.

3. The method of claim 1, in which the entry for the further multimedia object comprises the transmission fingerprint.

4. The method of claim 1, in which the entry for the further multimedia object comprises a fingerprint for an entry in the primary database matching the transmission fingerprint.

5. The method of claim 1, in which a transmission fingerprint is computed for plural further multimedia objects transmitted on respective transmission channels.

6. The method of claim 1, further comprising recording at least a portion of the obtained metadata and an identifier for the transmission channel in a transmission monitoring log-file.

7. The method of claim 1, further comprising removing a previous entry associated with a particular transmission channel from the secondary database upon adding the entry for the further multimedia object, associated with that particular transmission channel to the secondary database.

8. A system arranged for obtaining metadata for a multimedia object comprising matching means for matching an object fingerprint for the multimedia object against entries stored in a primary database, characterized by transmission monitoring means for recording a portion of a further multimedia object transmitted on a transmission channel and fingerprinting means for computing a transmission fingerprint for the portion, the matching means being arranged for matching the transmission fingerprint with the fingerprints stored in the primary database, adding an entry for the further multimedia object in a secondary database upon a successful match, and obtaining the metadata by matching the fingerprint against entries stored in the secondary database and only matching against the primary database if said matching against the secondary database fails.

9. The system of claim 8, further comprising receiving means for receiving at least a portion of the multimedia object, the fingerprinting means being arranged for computing the object fingerprint over the received portion.

10. A computer program product arranged for causing a processor to execute the method of claim 1.

\* \* \* \* \*

57/3,K/96 (Item 90 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0010839364  
WPI ACC NO: 2001-457387/  
XRPX Acc No: N2001-338987

Data compression process for use in storing and retrieving electronic data, uses two lists, the first points to entries in the second list that identify entries in a dictionary database of unique elements in the data to be stored

Patent Assignee: ARCHBOLD J (ARCH-I); MELE G (MELE-I); ZENTRONIX PTY LTD (ZENT-N)

Inventor: ARCHBOLD J; MELE G

Patent Family (6 patents, 92 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
WO 2001048593	A1	20010705	WO 2000AU1594	A	20001221	200149 B
AU 200123320	A	20010709	AU 200123320	A	20001221	200164 E
EP 1257900	A1	20021120	EP 2000986892	A	20001221	200301 E
			WO 2000AU1594	A	20001221	
US 20030093418	A1	20030515	WO 2000AU1594	A	20001221	200335 E
			US 2002168439	A	20021011	
JP 2003523564	W	20030805	WO 2000AU1594	A	20001221	200353 E
			JP 2001549179	A	20001221	
AU 777314	B2	20041014	AU 200123320	A	20001221	200501 E

21  
DEC  
2000  
Filing

Priority Applications (no., kind, date): AU 19994865 A 19991223

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 2001048593	A1	EN	23	0	

National Designated States, Original: AE AG AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW

Regional Designated States, Original: AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZW

AU 200123320 A EN Based on OPI patent WO 2001048593

EP 1257900 A1 EN PCT Application WO 2000AU1594

Based on OPI patent WO 2001048593

Regional Designated States, Original: AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI TR

US 20030093418 A1 EN PCT Application WO 2000AU1594

JP 2003523564 W JA 26 PCT Application WO 2000AU1594

Based on OPI patent WO 2001048593

AU 777314 B2 EN Previously issued patent AU 200123320

Based on OPI patent WO 2001048593 ..

Data compression process for use in storing and retrieving electronic data, uses two lists, the first points to entries in the second list that identify entries in a dictionary database of unique elements in the data to be stored

Alerting Abstract ...NOVELTY - Data is stored in the form of a dictionary database containing one copy of all repetitive elements found in the data



to be stored and two lists of location pointers . The first list contains pointers to locations in the second list which point to locations in the dictionary database , it may also include portions of text which are found only once in the data...

...ADVANTAGE - By using low value pointers , i.e. containing the least number of bits, to point to most often repeated elements of the stored data , the system can reduce storage needed to hold that data . Searching for data items may be improved, since the first list may be used when searching for elements in that stored data , i.e. searching for a pointer rather than having to continually convert data to and from its compressed form. Since the indexing system effectively replaces common data items with pointers which act as tokens, the system can be applied to data encryption/decryption...

Title Terms.../Index Terms/Additional Words: COMPRESS ; ...

... LIST ; ...

... DATABASE ;

#### Class Codes

International Classification (Main): G06F-017/30 ...

... G06F-007/00

(Additional/Secondary): G06F-012/00 ...

Manual Codes (EPI/S-X): T01-E ...

... T01-J05B2 ...

... T01-J05B4P ...

... T01-S03

#### Original Publication Data by Authority

#### Original Abstracts:

A method of storing data including the steps of providing a first index of first location identifiers , a second index of second location identifiers and a dictionary data base of data items, wherein the first location identifiers are adapted to identify the location of second location identifiers in the second index and the second location identifiers are adapted to identify the location of data items in the dictionary data base , receiving data and separating the data into a plurality of data items and storing the data items in a main data base , whereby at least one of the data items is stored in the main data base as at least one first location identifier , which identifies at least one second location identifier , which identifies the or each data item in the dictionary data base .

...

...A method of storing data including the steps of providing a first index of first location identifiers , a second index of second location identifiers and a dictionary data base of data items, wherein the first location identifiers are adapted to identify the location of second location identifiers in the second index and the second

**location identifiers** are adapted to identify the **location** of **data** items in the dictionary **data base**, **receiving data** and separating the **data** into a plurality of data items and storing the data items in a **main data base**, whereby at least one of the data items is stored in the **main data base** as at least one **first location identifier**, which identifies at least one second **location identifier**, which identifies the or each **data** item in the dictionary **data base**.

...

...A method of storing data including the steps of providing a first **index** of first **location identifiers**, a second **index** of **second location identifiers** and a dictionary **data base** of **data** items, wherein the **first location identifiers** are adapted to identify the **location** of **second location identifiers** in the second **index** and the second **location identifiers** are adapted to identify the **location** of **data** items in the dictionary **data base**, **receiving data** and separating the **data** into a plurality of data items and storing the data items in a **main data base**, whereby at least one of the data items is stored in the **main data base** as at least one **first location identifier**, which identifies at least one second **location identifier**, which identifies the or each **data** item in the dictionary **data base**.

...

...L'invention concerne un procede de stockage de donnees, consistant a fournir un premier **index** de premiers identificateurs d'emplacement, un second **index** de second identificateurs d'emplacement, et une base de donnees de dictionnaires d'articles de donnees, les premiers identificateurs d'emplacement etant concus pour **identifier** l'emplacement des seconds identificateurs d'emplacement dans le second **index**, et les seconds identificateurs d'emplacement etant concus pour **identifier** l'emplacement des articles de donnees dans la base de donnees de dictionnaires; a recevoir

**Claims:**

b 1 /b . A method of storing data including the steps of providing a first **index** of first **location identifiers**, a second **index** of **second location identifiers** and a dictionary **data base** of **data** items, wherein the **first location identifiers** are adapted to identify the **location** of **second location identifiers** in the second **index** and the second **location identifiers** are adapted to identify the **location** of **data** items in the dictionary **data base**, **receiving data** and separating the **data** into a plurality of data items and storing the data items in a **main data base**, whereby at least one of the data items is stored in the **main data base** as at least one **first location identifier**, which identifies at least one second **location identifier**, which identifies the or each **data** item in the dictionary **data base**.

57/3,K/112 (Item 106 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0010441658 - Drawing available  
WPI ACC NO: 2001-040672/200105  
XRPX Acc No: N2001-030337

**Application instantiation method for computer, involves extracting  
attribute value data from metabase repository and instantiating  
application layers residing in computer**

Patent Assignee: AMBROSE J (AMBR-I); ROTHWEIN T M (ROTH-I); SIEBEL  
SYSTEMS INC (SIEB-N); STAUBER C (STAU-I)

Inventor: AMBROSE J; ROTHWEIN T M; STAUBER C

**Patent Family** (7 patents, 89 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 2000052603	A1	20000908	WO 2000US5449	A	20000303	200105	B
AU 200033920	A	20000921	AU 200033920	A	20000303	200105	E
EP 1203310	A1	20020508	EP 2000912140	A	20000303	200238	E
			WO 2000US5449	A	20000303		
US 20020161734	A1	20021031	US 1999261771	A	19990303	200274	E
JP 2003505750	W	20030212	JP 2000602956	A	20000303	200321	E
			WO 2000US5449	A	20000303		
US 6574635	B2	20030603	US 1999261771	A	19990303	200339	E
US 20030120675	A1	20030626	US 1999261771	A	19990303	200343	E
			US 2003360269	A	20030206		

Priority Applications (no., kind, date): US 2003360269 A 20030206; US  
1999261771 A 19990303

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 2000052603	A1	EN	36	4	

National Designated States,Original: AE AL AM AT AU AZ BA BB BG BR BY CA  
CH CN CR CU CZ DE DK DM EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE  
KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU  
SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW

Regional Designated States,Original: AT BE CH CY DE DK EA ES FI FR GB GH  
GM GR IE IT KE LS LU MC MW NL OA PT SD SE SL SZ TZ UG ZW

AU 200033920 A EN Based on OPI patent WO 2000052603

EP 1203310 A1 EN PCT Application WO 2000US5449

Based on OPI patent WO 2000052603

Regional Designated States,Original: AL AT BE CH CY DE DK ES FI FR GB GR  
IE IT LI LT LU LV MC MK NL PT RO SE SI

JP 2003505750 W JA 43 PCT Application WO 2000US5449

Based on OPI patent WO 2000052603

US 20030120675 A1 EN Continuation of application US  
1999261771

**Application instantiation method for computer, involves extracting  
attribute value data from metabase repository and instantiating  
application layers residing in computer**

#### Original Titles:

...INSTANTIATION APPLICATION USING META DATA REPOSITORY

...

3  
MARCH  
1999

FILING

...APPLICATION INSTANTIATION BASED UPON ATTRIBUTES AND VALUES STORED IN A **META DATA REPOSITORY** , INCLUDING TIERING OF APPLICATION LAYERS OBJECTS AND COMPONENTS...

...Application instantiation based upon attributes and values stored in a **meta data repository** , including tiering of application layers, objects, and components...

...Application instantiation based upon attributes and values stored in a **meta data repository** , including tiering of application layers objects and components...

...INSTANTIATION APPLICATION USING **META DATA REPOSITORY**

**Alerting Abstract** ...Multiple layers contain objects and interfaces to pass data and commands between layers. Attribute value **data** is extracted from a **meta database repository** and the layer residing in the computer is instantiated after the extraction of the attribute...

...data manager layer and object manager layer. An INDEPENDENT CLAIM is also included for the **database** management system...

...ADVANTAGE - Allows companies to customize their applications and to **receive** the benefits of upgrades and supports. Provides internet based client/ **server** application capable of distributing itself widely across the internet and supporting multiple combinations and configurations. Improves performance and scalability and allows developers to create optimum sales, marketing and service **information** system configurations. Enables easier maintenance, **reduces** development time. Enables developers to assemble new applications and utilizes from existing software components with...

Title Terms.../Index Terms/Additional Words: **REPOSITORY** ;

Original Publication Data by Authority

**Original Abstracts:**

...in instantiation of multitiered applications having a user interface tier on the client, browser, or **remote** computer, from a meta data repository containing attributes and values of the attributes...

...in instantiation of multitiered applications having a user interface tier on the client, browser, or **remote** computer, from a **meta data repository** containing attributes and values of the attributes...

**Claims:**

...interfaces whereby to pass data and commands between layers, said method comprising extracting attribute-value **data** from a **metadatabase repository** and instantiating the layer residing on one of said computers  
...

...interfaces whereby to pass data and commands between layers, said method comprising extracting attribute-value **data** from a **metadatabase repository** and instantiating the layer residing on one of said computers  
...

...readable program code for establishing a multi-layered, object-oriented software application, comprising:an object **repository** containing a plurality of object attributes and definitions, along with associated

values; a **data** manager layer **connected** to the **object repository**, running on a **first** computer and operable to maintain an object-oriented abstraction of data within the **object repository**; an **object** manager layer **connected** to the **object repository** and the **data** manager layer, running on the first computer and operable to maintain a plurality of business...

...for communications between the layers, a user, or an external application, and further wherein the **object repository** contains an executable file for instantiating the user-interface objects in accordance with their...



57/3,K/57 (Item 51 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 The Thomson Corporation. All rts. reserv.

0013076060 - Drawing available

WPI ACC NO: 2003-156363/200315

Related WPI Acc No: 2002-566953; 2003-074756; 2003-090905; 2003-102825;

2003-102942; 2003-110871; 2003-174622; 2003-174624; 2003-198567;

2003-198602; 2003-198603; 2003-198604; 2003-199853; 2003-209442;

2003-238461; 2003-265798; 2003-266196; 2003-276087; 2003-276243;

2003-288711; 2005-505116; 2006-171305; 2006-442695

XRPX Acc No: N2003-123450

**Computer network file system has file server that selects and pushes infrequently accessed file to another file server**

Patent Assignee: BOLSTAD G D (BOLS-I); PRIESTER W G (PRIE-I); RANDALL J G (RAND-I); SCHWEITZER J R (SCHW-I); STAUB J R (STAU-I); ULRICH T R (ULRI-I)

Inventor: BOLSTAD G D; PRIESTER W G; RANDALL J G; SCHWEITZER J R; STAUB J R ; ULRICH T R

**Patent Family** (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20020161850	A1	20021031	US 2001264668	P	20010129	200315 B
			US 2001264669	P	20010129	
			US 2001264670	P	20010129	
			US 2001264671	P	20010129	
			US 2001264672	P	20010129	
			US 2001264673	P	20010129	
			US 2001264694	P	20010129	
			US 2001302424	P	20010629	
			US 200260977	A	20020129	

Priority Applications (no., kind, date): US 2001302424 P 20010629; US 2001264694 P 20010129; US 2001264673 P 20010129; US 2001264672 P 20010129; US 2001264671 P 20010129; US 2001264670 P 20010129; US 2001264669 P 20010129; US 2001264668 P 20010129; US 200260977 A 20020129

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20020161850	A1	EN	95	44	Related to Provisional US 2001264668
					Related to Provisional US 2001264669
					Related to Provisional US 2001264670
					Related to Provisional US 2001264671
					Related to Provisional US 2001264672
					Related to Provisional US 2001264673
					Related to Provisional US 2001264694
					Related to Provisional US 2001302424

**Computer network file system has file server that selects and pushes infrequently accessed file to another file server**

**Alerting Abstract ...NOVELTY** - A file server selects and transfers an infrequently accessed file to another file server . The file servers are loaded with file system meta data which allows a client (110) to locate files stored by the servers without prior knowledge...

**DESCRIPTION** - An INDEPENDENT CLAIM is included for data storage method

...

...ADVANTAGE - Since the infrequently accessed files are selected and pushed from one file **server** to **another** file **server**, the balance of unused storage capacity between the servers are improved without loading the **server**, thereby enabling the users to freely access any file without having specific knowledge of the file's current physical location. Hence, the **data storage** capacity and/or performance of the **data storage** system is increased without modifying the configuration of clients accessing the system...

...DESCRIPTION OF DRAWINGS - The figure shows the block diagram of a **server** node in the distributed file storage system...

#### Original Publication Data by Authority

#### Original Abstracts:

A data path controller architecture for a file **server** is described. The system includes a network interface for communicating with one or more clients and a storage interface for communicating with one or more disk drives. The file **server** further includes a data engine that is configured to communicate with the storage interface to **receive file data** from the one or more disk drives. The data engine is further configured to communicate with the network interface to **send file data** to the one or more clients. A CPU is configured to queue transaction requests for the data engine in response to file requests from the one or more clients. The **data** engine is configured to **receive file data** in response to at least a portion of the transaction requests. The **data** engine is further configured to **send file data** to the one or more clients in response to at least a portion of the...

#### Claims:

What is claimed is: b 1 /b . A computer network file system, comprising: a **first file server**; a **second file server**, said **first file server** configured to select an infrequently accessed file and push said infrequently accessed files to said **second server**, thereby improving a balance of unused storage capacity between the **first server** and the **second server** without substantially affecting a loading for each of the first and second servers; first **file system metadata** loaded on said **first file server**; and **second file system metadata** loaded on said **second file server**, said **first file system metadata** and said **second file system metadata** configured to allow a client to locate files stored by said **first file server** and files stored by said **second file server** without prior knowledge as to which **file server** stores said files.

57/3,K/54 (Item 48 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013116914 - Drawing available

WPI ACC NO: 2003-198603/200319

Related WPI Acc No: 2002-566953; 2003-074756; 2003-090905; 2003-102825;  
2003-102942; 2003-110871; 2003-156363; 2003-174622; 2003-174624;  
2003-198567; 2003-198602; 2003-198604; 2003-199853; 2003-209442;  
2003-238461; 2003-265798; 2003-266196; 2003-276087; 2003-276243;  
2003-288711; 2005-505116; 2006-171305; 2006-442695

XRPX Acc No: N2003-157825

Computer network file system for data storage and management system,  
has file server which mutually store attribute information sufficient to  
regenerate information in file system of another file server

Patent Assignee: BOLSTAD G D (BOLS-I); PRIESTER W G (PRIE-I); RANDALL J G  
(RAND-I); SCHWEITZER J R (SCHW-I); STAUB J R (STAU-I); ULRICH T R  
(ULRI-I)

Inventor: BOLSTAD G D; PRIESTER W G; RANDALL J G; SCHWEITZER J R; STAUB J R  
; ULRICH T R

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20020156974	A1	20021024	US 2001264668	P	20010129	200319 B
			US 2001264669	P	20010129	
			US 2001264670	P	20010129	
			US 2001264671	P	20010129	
			US 2001264672	P	20010129	
			US 2001264673	P	20010129	
			US 2001264694	P	20010129	
			US 2001302424	P	20010629	
			US 200260879	A	20020129	

Priority Applications (no., kind, date): US 2001302424 P 20010629; US  
2001264694 P 20010129; US 2001264673 P 20010129; US 2001264672 P  
20010129; US 2001264671 P 20010129; US 2001264670 P 20010129; US  
2001264669 P 20010129; US 2001264668 P 20010129; US 200260879 A  
20020129

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20020156974	A1	EN	96	44	Related to Provisional US 2001264668
					Related to Provisional US 2001264669
					Related to Provisional US 2001264670
					Related to Provisional US 2001264671
					Related to Provisional US 2001264672
					Related to Provisional US 2001264673
					Related to Provisional US 2001264694
					Related to Provisional US 2001302424

Computer network file system for data storage and management system,  
has file server which mutually store attribute information sufficient to  
regenerate information in file system of another file server

Alerting Abstract ...NOVELTY - A pair of file servers interconnected  
through a network store information specific to two different file  
systems. Each file server mutually stores attribute information that can

regenerate the information file system of the other **server** , when a client computer requests the associated file. DESCRIPTION - An INDEPENDENT CLAIM is included for **data storage** method...

...USE - For storing files including data such as video **data** , music **data** , news, etc., on **server** computers **connected** through network for **data storage** and management...

...Provides flexible and reliable storage of files and enables retrieval of files from a failed **server** by storing attribute information of one **server** in another **server** , thereby improving disk space utilization. The **file** servers allow client computer **connected** to the network to locate **files** owned by the file servers without prior knowledge as to which file **server** owns the files...

...DESCRIPTION OF DRAWINGS - The figure shows the block diagram of the **meta data** structures which are interlinked.

#### Original Publication Data by Authority

#### Original Abstracts:

...described. The redundant dynamically distributed file system operates on a computer network and includes a **first file server** that is operably **connected** to a network fabric and a second **file server** that is operably **connected** to the network fabric. The redundant dynamically distributed file system further includes first file system information loaded on the **first file server** and **second file system** information loaded on the **second file server** . The **first file system** information and the **second file system** information are configured to allow a client computer that is operably **connected** to the network fabric to locate **files** owned by the **first file server** and files owned by the **second file server** without prior knowledge as to which file **server** owns the files. The **first file server** is configured to store information sufficient to regenerate the **second file system** information, and the **second file server** is configured to store information sufficient to regenerate the first file system information.

#### Claims:

What is claimed is: b 1 /b . A computer network **file** system, comprising: a **first file server** operably **connected** to a network fabric; a second **file server** operably **connected** to the network fabric; **first file system information** loaded on said **first file server** ; and second file system information loaded on said second **file server** , said first file system information and said second file **system** information configured to allow a client computer **operably** connected to said network fabric to locate files owned by said first **file server** and files owned by said second **file server** without prior knowledge as to which **file server** owns said files, said first **file server** configured to store information sufficient to regenerate said second file system information, and said second **file server** configured to store information sufficient to regenerate said first file system information.

57/3,K/63 (Item 57 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0012872657 - Drawing available  
WPI ACC NO: 2002-731698/200279  
XRPX Acc No: N2002-576853

Informational database refreshing system for corporate database  
management system, has log monitor that dynamically analyzes log entries by  
using rule set that specifies selection criteria

Patent Assignee: ORACLE CORP (ORAC-N); ORACLE INT CORP (ORAC-N); REED D  
(REED-I)

Inventor: REED D

Patent Family (8 patents, 99 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20020128996	A1	20020912	US 2001804672	A	20010309	200279 B
WO 2002073465	A2	20020919	WO 2002US5935	A	20020225	200279 E
EP 1368753	A2	20031210	EP 2002707913	A	20020225	200382 E
			WO 2002US5935	A	20020225	
AU 2002242282	A1	20020924	AU 2002242282	A	20020225	200433 E
US 6832229	B2	20041214	US 2001804672	A	20010309	200501 E
CN 1535434	A	20041006	CN 2002806284	A	20020225	200506 E
JP 2005502934	W	20050127	JP 2002572052	A	20020225	200510 E
			WO 2002US5935	A	20020225	
IN 200301123	P2	20051014	WO 2002US5935	A	20020225	200580 E
			IN 2003KN1123	A	20030903	

Priority Applications (no., kind, date): US 2001804672 A 20010309

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20020128996	A1	EN	14	8	
WO 2002073465	A2	EN			

National Designated States,Original: AE AG AL AM AT AU AZ BA BB BG BR BY  
.BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID  
IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ  
NO NZ OM PH PL PT RO RU SD SE SG SI SK SL TJ TM TN TR TT TZ UA UG UZ VN  
YU ZA ZM ZW

Regional Designated States,Original: AT BE CH CY DE DK EA ES FI FR GB GH  
GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZM ZW

EP 1368753 A2 EN PCT Application WO 2002US5935  
Based on OPI patent WO 2002073465

Regional Designated States,Original: AL AT BE CH CY DE DK ES FI FR GB GR  
IE IT LI LT LU LV MC MK NL PT RO SE SI TR

AU 2002242282 A1 EN Based on OPI patent WO 2002073465  
JP 2005502934 W JA 41 PCT Application WO 2002US5935

Based on OPI patent WO 2002073465  
IN 200301123 P2 EN PCT Application WO 2002US5935

Informational database refreshing system for corporate database  
management system, has log monitor that dynamically analyzes log entries by  
using rule set that...

#### Original Titles:

...SYSTEM AND METHOD FOR MAINTAINING LARGE-GRAINED DATABASE CONCURRENCY  
WITH A LOG MONITOR INCORPORATING DYNAMICALLY REDEFINABLE BUSINESS LOGIC...



...System and method for maintaining large-grained **database** concurrency with a log monitor incorporating dynamically redefinable business logic...

...System and method for maintaining large-grained **database** concurrency with a log monitor incorporating dynamically redefinable business logic...

...SYSTEM AND METHOD FOR MAINTAINING LARGE-GRAINED **DATABASE** CONCURRENCY WITH A LOG MONITOR INCORPORATING DYNAMICALLY REDEFINABLE BUSINESS LOGIC...

**Alerting Abstract** ...NOVELTY - A production **database** (13) stores records of production data generated by a transaction processing system. A log monitor (26) dynamically analyzes the log entries generated for each transaction committed to the **database** by using rule set that specifies a selection criteria. The **updated** records of the production **data** satisfying the selection criteria are stored into an informational **database** . . . **Informational database refreshing** method; Computer-readable storage medium storing **informational database refreshing** program; System for maintaining large-grained **database** concurrence with log monitor incorporating dynamically redefined business logic; Method for maintaining large-grained **database** concurrence with log monitor incorporating dynamically redefined business logic; and Computer readable storage medium storing large-grained **database** concurrence maintenance program...

...USE - **Information database refreshing** system for corporate **database** management system for on-line transaction processing systems and e-commerce systems...

...ADVANTAGE - The system non-intrusively **updates** an **informational database** with minimal effect on production system operation and autonomous operations. The resource expenditures are **reduced** by avoiding **data** duplication and inefficient data retrieval. The business logic of selected **data** is redefined in flexible manner by **updating** the **informational database** with high frequency and low overhead...

...OF DRAWINGS - The figure shows the block diagram of the computer environment that incorporates the **informational database refreshing** system...

...13 Production **database**

**Title Terms/Index Terms/Additional Words:** **DATABASE** ; ...

... **REFRESH** ;

**Original Publication Data by Authority**

**Original Abstracts:**

A system (10) and method (150) for maintaining large-grained **database** concurrency with a log monitor (26) incorporating dynamically redefinable business logic (94) are described. Operations expressed in a data manipulation language are executed against a **source database** (51). At least one operation constitutes a commit operation that completes each **database** transaction. A current rule set (55) is defined. Each rule

includes business logic (94) specifying a data selection criteria for records stored in the **source database** . A log entry (70) is periodically generated in a log (54) for each transaction committed to the **source database** (51). Each log entry (70) identifies an affected record and includes transactional data. The transaction...

...in the current rule set (55). A new record (57) is built in accordance with **metadata** (56) describing a **destination database** (58). The new record (57) contains select transactional data from the log entry (70) of each transaction meeting the selection criteria. The new record (57) is stored into the **destination database** (58). The data stored in the **destination database** (58) includes at least a partial subset of the **source database** (51)...

...A system and method for maintaining large-grained **database** concurrency with a log monitor incorporating dynamically redefinable business logic are described. Operations expressed in a data manipulation language are executed against a **source database** . At least one operation constitutes a commit operation that completes each **database** transaction. A current rule set is defined. Each rule includes business logic specifying a data selection criteria for records stored in the **source database** . A log entry is periodically generated in a log for each transaction committed to the **source database** . Each log entry identifies an affected record and includes transactional data. The transaction identified in...

...criteria specified in the current rule set. A new record is built in accordance with **metadata** describing a **destination database** . The new record contains select transactional data from the log entry of each transaction meeting the selection criteria. The new record is stored into the **destination database** . The data stored in the **destination database** includes at least a partial subset of the **source database** .

...

...A system and method for maintaining large-grained **database** concurrency with a log monitor incorporating dynamically redefinable business logic are described. Operations expressed in a data manipulation language are executed against a **source database** . At least one operation constitutes a commit operation that completes each **database** transaction. A current rule set is defined. Each rule includes business logic specifying a data selection criteria for records stored in the **source database** . A log entry is periodically generated in a log for each transaction committed to the **source database** . Each log entry identifies an affected record and includes transactional data. The transaction identified in...

...criteria specified in the current rule set. A new record is built in accordance with **metadata** describing a **destination database** . The new record contains select transactional data from the log entry of each transaction meeting the selection criteria. The new record is stored into the **destination database** . The data stored in the **destination database** includes at least a partial subset of the **source database** .

...

...A system (10) and method (150) for maintaining large-grained **database** concurrency with a log monitor (26) incorporating dynamically redefinable business logic (94) are described. Operations expressed in a data

manipulation language are executed against a **source database** (51). At least one operation constitutes a commit operation that completes each **database** transaction. A current rule set (55) is defined. Each rule includes business logic (94) specifying a data selection criteria for records stored in the stored in the **source database** . A log entry (70) is periodically generated in a log (54) for each transaction committed to the **source database** (51). Each log entry (70) identifies an affected record and includes transactional data. The transaction...

...in the current rule set (55). A new record (57) is built in accordance with **metadata** (56) describing a **destination database** (58). The new record (57) contains select transactional data from the log entry (70) of each transaction meeting the selection criteria. The new record (57) is stored into the **destination database** (58). The data stored in the **destination database** (58) includes at least a partial subset of the **source database** (51

**Claims:**

What is claimed is: b 1 /b . A system for **refreshing an informational database** through log-based transaction monitoring, comprising: a production **database** comprising one or more tables each storing records of production data generated by a transaction...

...file with at least one log entry generated for each transaction committed to the production **database** ; an informational **database** comprising one or more tables each storing records of informational data for use by a...

...analyzing the log entries stored into the log file using a rule set that specifies a data selection criteria and **storing** updated records generated from **production** data satisfying the data selection criteria into the **informational database**...

...What is claimed is: 1. A method for using a log associated with a **first database** to update a **second database**, the method comprising the computer-implemented steps of: based on said log that is associated with **said first database**, **identifying** first data; **generating second data** based on **said first data**; and **sending** said second data to **said second database**.

57/3,K/81 (Item 75 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0012265474 - Drawing available  
WPI ACC NO: 2002-205680/200226  
XRPX Acc No: N2002-156649

**System for accessing data in disparate information sources by maintaining relationships between physical metal-data elements in order to initiate data query requests**

Patent Assignee: HAUCH R (HAUC-I); METAMATRIX INC (META-N); MILLER A (MILL-I); SCANLON R (SCAN-I); WOLFANGEL S (WOLF-I); WRIGHT B (WRIG-I)  
Inventor: HAUCH R; MILLER A; MILLER A M; SCANLON R; WOLFANGEL S; WRIGHT B  
Patent Family (4 patents, 91 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
WO 2001075679	A1	20011011	WO 2001US10943	A	20010404	200226 B
AU 200153136	A	20011015	AU 200153136	A	20010404	200226 E
EP 1277138	A1	20030122	EP 2001926614	A	20010404	200308 E
			WO 2001US10943	A	20010404	
US 20040128276	A1	20040701	WO 2001US10943	A	20010404	200444 E
			US 2004450581	A	20040209	

Priority Applications (no., kind, date): US 2004450581 A 20040209; US 2000194925 P 20000404

**Patent Details**

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 2001075679	A1	EN	63	18	
National Designated States,Original: AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CO CR CU CZ DE DK DM EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG US UZ VN YU ZA ZW					
Regional Designated States,Original: AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZW					
AU 200153136	A	EN			Based on OPI patent WO 2001075679
EP 1277138	A1	EN			PCT Application WO 2001US10943
					Based on OPI patent WO 2001075679
Regional Designated States,Original: AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI TR					
US 20040128276	A1	EN			PCT Application WO 2001US10943

**Alerting Abstract ...NOVELTY** - A **meta** -matrix **server** (100) is coupled to information **sources** (130a-130e) and to a **meta** -matrix **repository** (110) comprising a facility (110a) communicating data to the **server** and a **repository** (110b) storing the various **meta** -models, while a user may select a run-time **meta** -model from **repository** (110b) which is then employed to access information stored in the various **information** sources corresponding to the **meta** -model.DESCRPTION - A modeler (110c) **imports data** from the sources and generates **meta** -models...

...100 **Server**

...

...110 **Repositories**

## Original Publication Data by Authority

### Original Abstracts:

The present invention relates to a system (10) for generating and maintaining virtual and physical **metadata** layers in a MetaBase **metadata repository** (110b) in order to simplify and optimize the retrieval of data from a plurality of disparate information sources (130a-130c). The system stores in a physical **metadata** layer of a MetaBase **metadata repository** a plurality of physical **metadata** elements, wherein each one of the physical **metadata** elements corresponds to the **metadata** elements in the plurality of **information** sources. Logical **metadata** elements are stored in the virtual **metadata** layer and are **linked** to the physical **metadata** elements in order to maintain the relationships therebetween. By maintaining the relationships between the physical **metadata** elements, users can initiate a **data** query request for **data** corresponding to a logical **metadata** element, and the system is configured to retrieve the desired data from the relevant information sources, even in the event that relevant information **sources** maintain the data in fields having **different data field names**, that the **information sources** employ incompatible data formats, and that the relevant information sources employ different query languages...

...The present invention relates to a system ( b 10 /b ) for generating and maintaining virtual and physical **metadata** layers in a MetaBase **metadata repository** ( b 110 /b b) in order to simplify and optimize the retrieval of data from a plurality of disparate information sources ( b 130 /b a- b 130 /b c). The system stores in a physical metadata layer of a MetaBase metadata repository a **plurality** of physical metadata elements, wherein each one of the physical metadata elements **corresponds** to the metadata elements in the plurality of **information** sources. Logical metadata elements are stored in the virtual metadata layer and are linked to the physical metadata elements in order to maintain the relationships therebetween. By maintaining the relationships **between** the physical metadata elements, users **can** initiate a data **query** request for data corresponding to a logical metadata element, and the system is configured to retrieve the desired data from the relevant information sources, even in the event **that** relevant information sources maintain the data in fields having **different data field names**, that the information sources employ incompatible data formats, and that the relevant information sources employ different...

...The present invention relates to a system (10) for generating and maintaining **virtual** and physical metadata layers in a MetaBase metadata repository (110b) in order to simplify and optimize the retrieval of data from a plurality of disparate information sources (130a-130c). The system stores in a physical metadata layer of a MetaBase metadata repository a **plurality** of physical metadata elements, wherein each one of the physical metadata elements **corresponds** to the metadata elements in the plurality of **information** sources. Logical metadata elements are stored in the virtual metadata layer and are linked to the physical metadata elements in order to maintain the relationships therebetween. By maintaining the relationships **between** the physical metadata elements, users **can** initiate a data **query** request for data corresponding to a logical metadata element, and the system is configured to retrieve the desired data from the relevant information sources, even in the event **that** relevant information sources maintain the data in fields having



different data field names , that the information sources employ incompatible data formats, and that the relevant information sources employ different...

**Claims:**

...an information source having storage spaces, each said storage space configured to store types of **data** defined by physical **metadata** elements; a **metadata repository** configured to store two sets of **metadata** elements, wherein a first set of said **metadata** elements comprises said physical **metadata** elements of said **information** source, and wherein said second set of **metadata** elements comprises logical **metadata** elements, each of which correspond to at least one physical **metadata** element of said **first** set; and a **MetaMatrix** server coupled to said **metadata** repository and to said information source, wherein said **MetaMatrix** server is configured to receive a query request for a **logical** metadata element from a user via an application programming interface, and to retrieve from said information source **the** data defined by the corresponding **physical** metadata element.

57/3,K/28 (Item 22 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013802036 - Drawing available  
WPI ACC NO: 2003-902146/200382  
XRPX Acc No: N2003-720448

**Functional client side data cache coherence maintaining system, has client coupled to local memory cache and remote memory storage system to store, process, retrieve and transmit requested data to client**

Patent Assignee: ARNOLD R (ARNO-I); JACOB S (JACO-I); LEWIS K (LEWI-I);  
PERVASIVE SOFTWARE INC (PERV-N); REINAUER R (REIN-I); SUN C (SUNC-I);  
TAN D (TAND-I); WHITE K (WHIT-I)

Inventor: ARNOLD R; JACOB S; LEWIS K; REINAUER R; SUN C; TAN D; WHITE K

**Patent Family** (2 patents, 1 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
US 20030217081	A1	20031120	US 2002144917	A	20020514	200382 B
US 6996584	B2	20060207	US 2002144917	A	20020514	200611 E

Priority Applications (no., kind, date): US 2002144917 A 20020514

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing	Notes
US 20030217081	A1	EN	21	9		

**...coupled to local memory cache and remote memory storage system to store, process, retrieve and transmit requested data to client**

**Alerting Abstract** ...a communication pathway. The storage system stores, processes requests, retrieves from the storage system and **transmits** the requested **data** to the client with annotated version information.... included for a method of maintaining a functional client side data cache coherence distributed across **database** servers and clients...

...USE - Used for maintaining a functional coherent **database** cache distributed across both **database** **server** and client...

...system maintains client side data caches for read operations while maintaining functional coherence with a **server** side data cache, thereby allowing remote clients to access recently accessed, **server** based data **locally** without any required network communication or **server** interaction. The system has a **database** client cache synchronization mechanism, hence **reducing** unpredictable and erroneous application behavior as the clients attempt to collaborate...

...:30 **Server**

...  
... 42 **Meta data**

**Title Terms...**/Index Terms/Additional Words: **TRANSMIT ;**

**Original Publication Data by Authority**

#### Original Abstracts:

The present invention provides functional client side data cache coherence

distributed across **database** servers and clients. This system includes an application resident on a client operable to request...

...for specified data, retrieve the specified data from within the remote memory storage system, and **transmit** the requested **data** to the client with annotated version **information**. The **data received** by the client is verified as being coherent with any downstream **linked information** stored in the client's local memory cache. Otherwise, **updated** coherent **data** is requested, **received** and verified prior to being used by the client and its resident applications...

...The present invention provides functional client side data cache coherence distributed across **database** servers and clients. This system includes an application resident on a client operable to request...

...for specified data, retrieve the specified data from within the remote memory storage system, and **transmit** the requested **data** to the client with annotated version **information**. The **data received** by the client is verified as being coherent with any downstream **linked information** stored in the client's local memory cache. Otherwise, **updated** coherent **data** is requested, **received** and verified prior to being used by the client and its resident applications.

**Claims:**

...claimed is: b 1 /b . A system for maintaining functional client side data cache coherence distributed across **database** servers and clients, comprising:an application resident on a client operable to request access to...

...wherein said remote memory storage system is operable to store at least one piece of **stored . data** , receive a request for at least one piece of specified data, retrieve said at least one piece of specified data from said at least one piece of **stored data**, **and** transmit said at least one piece of **specified** data to said client, and wherein said at least one piece of specified data provided...

...What is claimed is:1. A system for maintaining functional client side data cache coherence **for** database caches distributed **across** database servers and clients, comprising: a database file comprising a plurality **of data** records including one or **more** linkages **between data** records;an application resident on a database client operable to request access to at least one piece of specified data within a record within **the** database file, and wherein said client is coupled to a local memory cache, wherein said local memory cache is operable to store a plurality **of data** records and version information related to the plurality **of data** records, the version information comprising first values representing versions related to **the data** records **and** second value representing versions related to **the** database file, wherein the application is configured to utilize the version information to determine **whether** locally **cached data** records have **valid** links with respect to each other; **and** a remote memory storage system configured to **manage** the database file and coupled to said client by a communication pathway, wherein said remote memory storage system is **operable** to receive the request for at least one piece of specified data, to **retrieve the data** record for the specified data **from** the database file **or** a remote cache, **and to transmit the data** record and associated version information; **wherein** the database file further comprises a plurality of page, each page including portion of one **or more** data records, and wherein version information is associated with each page individually;wherein said

application verifies that **any** page received from the remote memory storage system is **coherent with** linked data stored in pages within said local memory cache;wherein said application **uses** version information of both **the** page received **and** said linked pages stored in said local memory cache;wherein said version information represents a...

...said version range of said pages overlap; andwherein said version range comprises a version **for** when a linkage to or from **a** page was **updated** and a file version for when **said** page was downloaded to the local cache.



US 20030217081A1

(19) **United States**

(12) **Patent Application Publication**

White et al.

(10) **Pub. No.: US 2003/0217081 A1**

(43) **Pub. Date: Nov. 20, 2003**

(54) **SYSTEM AND METHOD OF MAINTAINING  
FUNCTIONAL CLIENT SIDE DATA CACHE  
COHERENCE**

(52) **U.S. Cl. .... 707/203**

(76) **Inventors: Ken White, Cedar Park, TX (US); Rob  
Reinauer, Austin, TX (US);  
Chunsheng Sun, Round Rock, TX  
(US); Richard Arnold, Austin, TX  
(US); Sunil Jacob, Austin, TX (US);  
Desmond Tan, Austin, TX (US); Kevin  
Lewis, Austin, TX (US)**

(57) **ABSTRACT**

The present invention provides functional client side data cache coherence distributed across database servers and clients. This system includes an application resident on a client operable to request access to data, and wherein the client is coupled to a local memory cache operable to store requested data. The client is coupled to a remote memory storage system, such as disk storage or network resources by a communication pathway. This remote memory storage system is operable to store data, process requests for specified data, retrieve the specified data from within the remote memory storage system, and transmit the requested data to the client with annotated version information. The data received by the client is verified as being coherent with any downstream linked information stored in the client's local memory cache. Otherwise, updated coherent data is requested, received and verified prior to being used by the client and its resident applications.

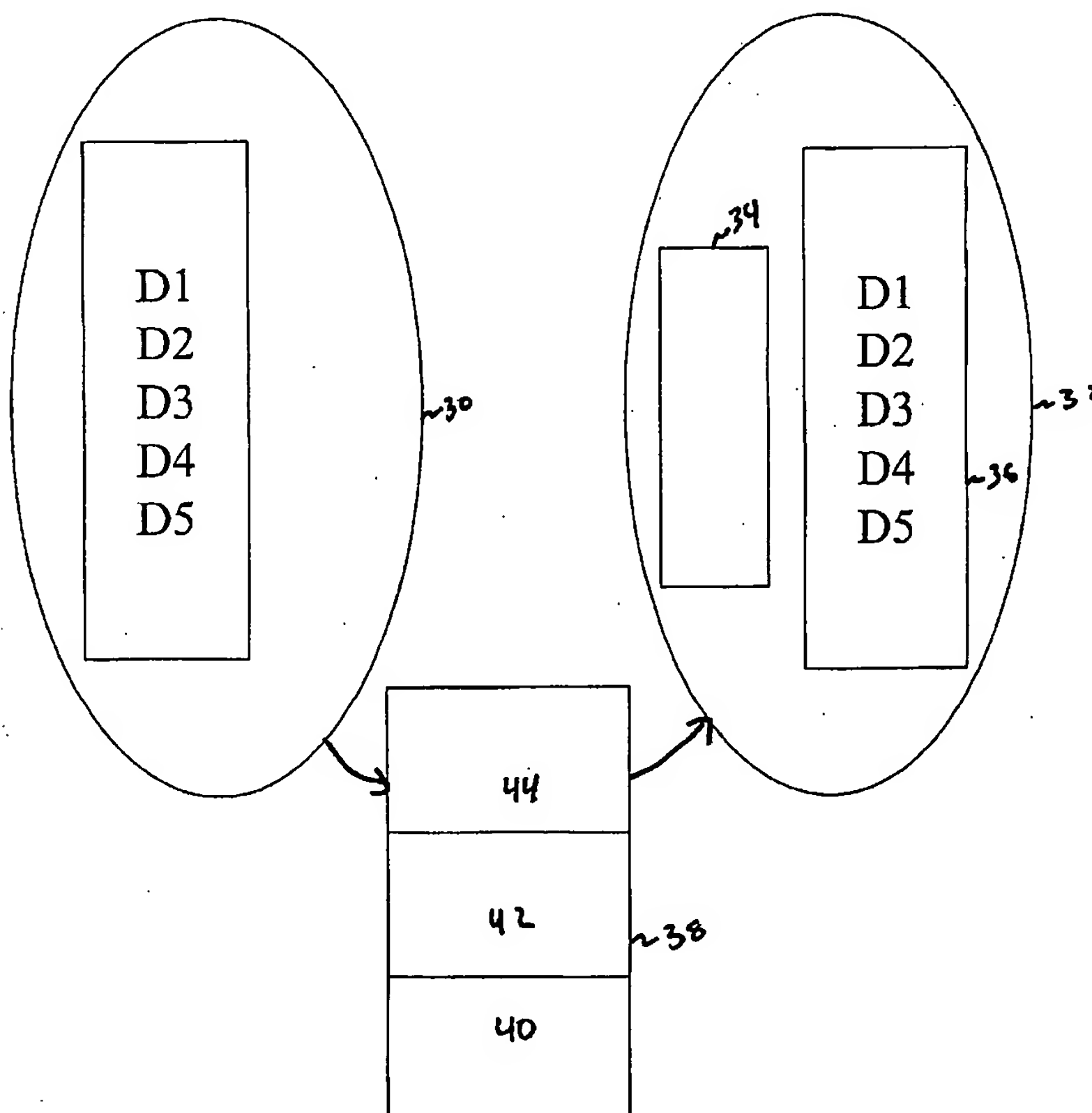
Correspondence Address:  
**HUGHES & LUCE LLP  
1717 MAIN STREET  
SUITE 2800  
DALLAS, TX 75201 (US)**

(21) **Appl. No.: 10/144,917**

(22) **Filed: May 14, 2002**

**Publication Classification**

(51) **Int. Cl.<sup>7</sup> ..... G06F 17/30**



[0128] One downside of this approach is that the client can only compare against the latest file version it has. If the client becomes fully cached with its data of interest in a short time span, the client will have no reason to invalidate any pages unless an automatic invalidation mechanism kicks in. If this occurs, the client may invalidate pages unnecessarily. This could be combined with a request for invalidated pages to reduce unnecessary page requests.

[0129] Another negative is that the file version does not represent all changes to the data, but those changes that affect linkages. Many changes may be made to non-indexed fields and this file version would remain the same.

[0130] The current embodiment uses a combination of active and passive invalidation. The server keeps a list of changed pages for each client, and the invalidations for a client are added to the response to a request from that client. In this way, the present invention takes advantage of existing network traffic, reducing network overhead (as in the passive approach). However, to guarantee deterministic latency, a client will make invalidation requests to the server if too much time has passed since a passive (i.e. piggybacked) invalidation request. This active invalidation is only used when there is no other traffic between the client and the server and does so in a time-based manner that is determined by the client's configuration." This could either replace or augment all of the invalidation discussion, which was really discovery & not design. The server has knowledge of the frequency with which changes are being introduced to the system. If the server is able to relay this information back to the client, the client can adjust its invalidation mechanism for optimum efficiency based on the state of the system overall. This could be pulled either at the time of page requests or during invalidation requests (if that is implemented).

[0131] Although the present invention has been described in detail, it should be understood that various changes, substitutions and alterations can be made hereto without departing from the spirit and scope of the invention as described by the appended claims.

What is claimed is:

1. A system for maintaining functional client side data cache coherence distributed across database servers and clients, comprising:

an application resident on a client operable to request access to at least one piece of specified data, and wherein said client is coupled to a local memory cache, wherein said local memory cache is operable to store said at least one piece of specified data; and

a remote memory storage system coupled to said client by a communication pathway, wherein said remote memory storage system is operable to store at least one piece of stored data, receive a request for at least one piece of specified data, retrieve said at least one piece of specified data from said at least one piece of stored data, and transmit said at least one piece of specified data to said client, and wherein said at least one piece of specified data provided to said client is annotated with version information.

2. The system for maintaining functional client side data cache coherence distributed across database servers and

clients of claim 1, wherein said at least one piece of specified data comprises a page, node, or linkage.

3. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 2, wherein a data file used by said application comprises said at least one piece of specified data.

4. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein said application verifies that any piece of specified data is coherent with linked data stored in said local memory cache.

5. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 4, wherein said application uses said version information of both said received at least one piece of specified data and said linked data stored in said local memory cache.

6. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein said version information comprises a version range and wherein any two pieces of data are coherent when said version range of said two pieces of data overlap.

7. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 6, wherein said version range comprises a file version for when a linkage to or from a node was updated and a file version for when said node was downloaded to a clients cache.

8. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 7, wherein said node comprises data, pages or links.

9. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein any writes to said data stored in either said local memory cache or said remote memory storage system are processed by said remote memory storage system prior to executing said write.

10. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein said specified data transmitted from said remote memory storage system is transmitted as part of a data payload, wherein said payload comprises normal data, invalidation information and an invalidation header.

11. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 10, wherein said normal data comprises said specified data annotated with version information.

12. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 11, wherein said invalidation header is synchronous or asynchronous.

13. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 11, wherein invalidations having synchronous invalidation headers are processed first.

14. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 11, wherein if said invalidation header is asynchronous, said invalidation is placed in an invalidation queue to be processed.



15. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 14, wherein a user specifies a frequency for processing said invalidation queue.

16. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein said remote memory storage system is a local disk accessed memory system.

17. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein said remote memory storage system is a network accessed resource.

18. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein said client and said remote memory storage system can switch to a client-server mode for improved efficiency when needed.

19. The system for maintaining functional client side data cache coherence distributed across database servers and clients of claim 1, wherein storing said at least one piece of stored data is tuned.

20. A method for maintaining functional client side data cache coherence distributed across database servers and clients, comprising:

requesting access to at least one piece of specified data from an application resident on a client, and wherein said client is coupled to a local memory cache;

coupling said client to a remote memory storage system, wherein said remote memory storage system is operable to:

store at least one piece of stored data;

receive a request for at least one piece of specified data from said client;

retrieve said at least one piece of specified data from said at least one piece of stored data; and

transmit said at least one piece of specified data to said client, and wherein said at least one piece of specified data provided to said client is annotated with version information;

verifying said at least one piece of specified data is coherent with any downstream linked information stored at said client in said local memory cache; and

storing said at least one piece of specified data in said local memory cache.

21. The method of claim 20, said data comprises a page, node, or linkage.

22. The method of claim 21, wherein a data file used by said application comprises said data.

23. The method of claim 20, wherein said application uses said version information of both said received at least one piece of specified data and said linked data stored in said local memory cache to verify coherence.

24. The method of claim 21, wherein said version information comprises a version range and wherein two pieces of data are coherent when said version range of said two pieces of data overlap.

25. The method of claim 24, wherein said version range comprises wherein said version range comprises a file version for when a linkage to or from a node was updated and a file version for when said node was downloaded to a clients cache.

26. The method of claim 20, wherein any writes to said data stored in either said local memory cache or said remote memory storage system are processed by said remote memory storage system prior to executing said write.

27. The method of claim 20, wherein data transmitted from said remote memory storage system comprises a data payload, wherein said payload comprises header information, invalidation meta data, and normal data.

28. The method of claim 27, wherein said normal data comprises said specified data annotated with version information.

29. The method of claim 27, wherein said invalidation meta data comprises a synchronous or asynchronous header.

30. The method of claim 28, wherein invalidations having synchronous invalidation headers are processed first.

31. The method of claim 29, wherein if said invalidation header is asynchronous, said invalidation is placed in an invalidation queue to be processed.

32. The method of claim 31, wherein a user specifies a frequency for processing said invalidation queue.

33. The method of claim 20, wherein said remote memory storage system is a local disk accessed memory system.

34. The method of claim 20, wherein said remote memory storage system is a network accessed resource.

\* \* \* \* \*



US006996584B2

(12) **United States Patent**  
**White et al.**

(10) **Patent No.:** **US 6,996,584 B2**  
(45) **Date of Patent:** **Feb. 7, 2006**

(54) **SYSTEM AND METHOD OF MAINTAINING  
FUNCTIONAL CLIENT SIDE DATA CACHE  
COHERENCE**

(75) **Inventors:** Ken White, Cedar Park, TX (US); Rob Reinauer, Austin, TX (US); Chunsheng Sun, Round Rock, TX (US); Richard Arnold, Austin, TX (US); Sunil Jacob, Austin, TX (US); Desmond Tan, Austin, TX (US); Kevin Lewis, Austin, TX (US)

(73) **Assignee:** Pervasive Software, Inc., Austin, TX (US)

(\*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 353 days.

(21) **Appl. No.:** 10/144,917

(22) **Filed:** May 14, 2002

(65) **Prior Publication Data**

US 2003/0217081 A1 Nov. 20, 2003

(51) **Int. Cl.**

G06F 17/30 (2006.01)

G06F 17/00 (2006.01)

G06F 15/16 (2006.01)

(52) **U.S. Cl.** ..... 707/203; 707/8; 707/100; 709/203

(58) **Field of Classification Search** ..... 707/1-4, 707/8, 10, 100-102, 104.1, 200-205; 709/201, 709/203, 217-219, 224, 226; 711/118, 141, 711/144, 154

See application file for complete search history.

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

5,452,447 A 9/1995 Nelson et al. .... 395/650  
5,581,704 A 12/1996 Barbara et al. .... 395/200.09

5,706,435 A 1/1998 Barbara et al. .... 395/200.09  
5,754,774 A 5/1998 Bittinger et al. .... 395/200.33  
5,907,678 A 5/1999 Housel, III et al. .... 398/200.43  
5,918,244 A 6/1999 Percival ..... 711/119  
6,032,228 A 2/2000 Islam et al. .... 711/133  
6,154,811 A 11/2000 Srbljic et al. .... 711/118  
6,219,676 B1 4/2001 Reiner ..... 707/201  
6,266,678 B1 7/2001 McDevitt et al. .... 707/201

(Continued)

**OTHER PUBLICATIONS**

Godlin et al., Method and System for Differential Distributed Data File Storage, Management and Access, Oct. 31, 2002.\*

(Continued)

*Primary Examiner*—Jean M. Corrielus

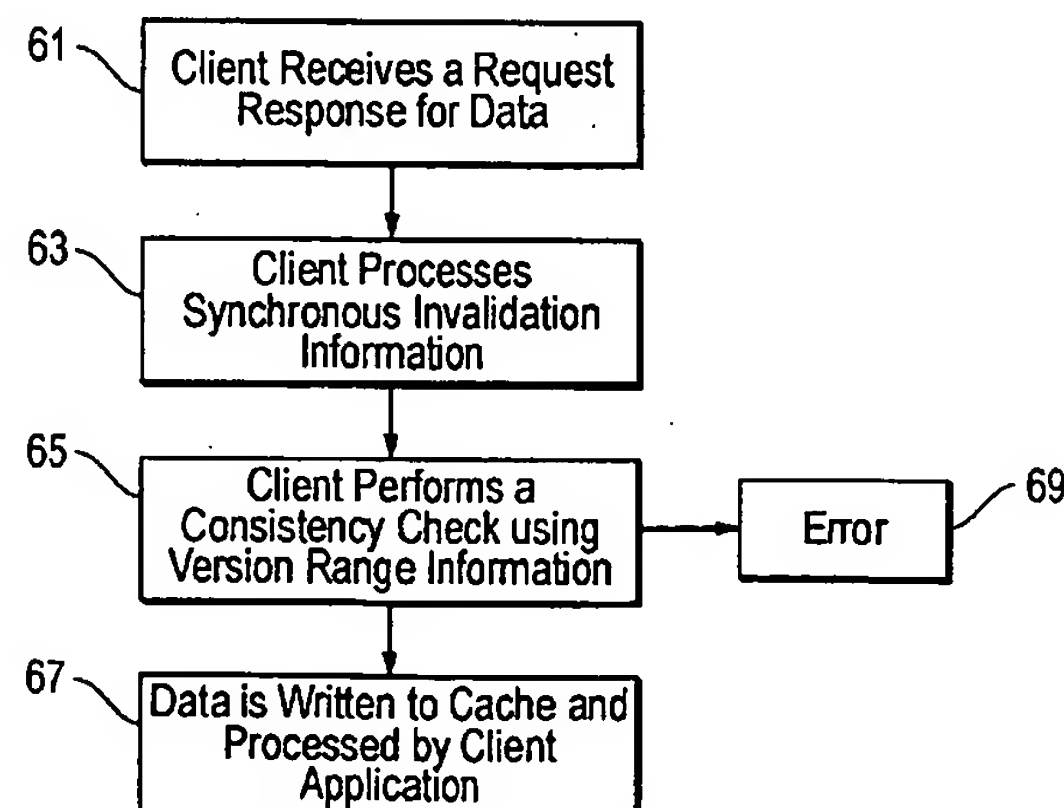
*Assistant Examiner*—Miranda Le

(74) *Attorney, Agent, or Firm*—Brian W. Peterman; O'keefe, Egan & Peterman, LLP

(57) **ABSTRACT**

The present invention provides functional client side data cache coherence distributed across database servers and clients. This system includes an application resident on a client operable to request access to data, and wherein the client is coupled to a local memory cache operable to store requested data. The client is coupled to a remote memory storage system, such as disk storage or network resources by a communication pathway. This remote memory storage system is operable to store data, process requests for specified data, retrieve the specified data from within the remote memory storage system, and transmit the requested data to the client with annotated version information. The data received by the client is verified as being coherent with any downstream linked information stored in the client's local memory cache. Otherwise, updated coherent data is requested, received and verified prior to being used by the client and its resident applications.

**24 Claims, 6 Drawing Sheets**



quently, it is more likely that the first write will be rejected, but any burst writing after that point is more likely to contain fresh data.

It is possible that the client cache has some timing pattern where it requests invalidations every so often whether an application is actively forcing server requests or not. This aspect could be bypassed, and could be tuned if the timing described above was in place. The client has control of the frequency of invalidation requests since the client makes this decision.

One downside of this is that the server is required to keep track of page changes that need to be sent to each client. This does not necessarily require multiple lists: the server could keep one list per file that was ordered in time and keep a marker for each client that will request invalidations (i.e., clients where invalidations are not bypassed) as to where the client was on the list. As clients move further in time down the list, the earlier list items can be released. This management does not have to be synchronous.

An additional negative is that this will increase network traffic. Since an increase in network data traffic likely requires an increase in network invalidation traffic, data and invalidations can form race conditions, with each contending for limited network bandwidth.

Since the client has some indication of the range of commits that are represented by the cached data, it can invalidate pages that are guaranteed to be consistent, but that it feels may be too old based on the current version of the file. This determination can be tunable at the client and can be influenced automatically by other factors (such as server state).

One downside of this approach is that the client can only compare against the latest file version it has. If the client becomes fully cached with its data of interest in a short time span, the client will have no reason to invalidate any pages unless an automatic invalidation mechanism kicks in. If this occurs, the client may invalidate pages unnecessarily. This could be combined with a request for invalidated pages to reduce unnecessary page requests.

Another negative is that the file version does not represent all changes to the data, but those changes that affect linkages. Many changes may be made to non-indexed fields and this file version would remain the same.

The current embodiment uses a combination of active and passive invalidation. The server keeps a list of changed pages for each client, and the invalidations for a client are added to the response to a request from that client. In this way, the present invention takes advantage of existing network traffic, reducing network overhead (as in the passive approach). However, to guarantee deterministic latency, a client will make invalidation requests to the server if too much time has passed since a passive (i.e. piggybacked) invalidation request. This active invalidation is only used when there is no other traffic between the client and the server and does so in a time-based manner that is determined by the client's configuration." This could either replace or augment all of the invalidation discussion, which was really discovery & not design. The server has knowledge of the frequency with which changes are being introduced to the system. If the server is able to relay this information back to the client, the client can adjust its invalidation mechanism for optimum efficiency based on the state of the system overall. This could be pulled either at the time of page requests or during invalidation requests (if that is implemented).

Although the present invention has been described in detail, it should be understood that various changes, substi-

tutions and alterations can be made hereto without departing from the spirit and scope of the invention as described by the appended claims.

What is claimed is:

1. A system for maintaining functional client side data cache coherence for database caches distributed across database servers and clients, comprising:

a database file comprising a plurality of data records including one or more linkages between data records; an application resident on a database client operable to request access to at least one piece of specified data within a record within the database file, and wherein said client is coupled to a local memory cache, wherein said local memory cache is operable to store a plurality of data records and version information related to the plurality of data records, the version information comprising first values representing versions related to the data records and second value representing versions related to the database file, wherein the application is configured to utilize the version information to determine whether locally cached data records have valid links with respect to each other; and

a remote memory storage system configured to manage the database file and coupled to said client by a communication pathway, wherein said remote memory storage system is operable to receive the request for at least one piece of specified data, to retrieve the data record for the specified data from the database file or a remote cache, and to transmit the data record and associated version information;

wherein the database file further comprises a plurality of page, each page including portion of one or more data records, and wherein version information is associated with each page individually;

wherein said application verifies that any page received from the remote memory storage system is coherent with linked data stored in pages within said local memory cache;

wherein said application uses version information of both the page received and said linked pages stored in said local memory cache;

wherein said version information represents a version range and wherein any two pages are coherent when said version range of said pages overlap; and

wherein said version range comprises a version for when a linkage to or from a page was updated and a file version for when said page was downloaded to the local cache.

2. The system of claim 1, wherein version information is associated with each data record individually.

3. The system of claim 1, wherein any writes to data stored in either said local memory cache or said remote memory storage system are processed by said remote memory storage system prior to executing said write.

4. The system of claim 1, wherein said specified data transmitted from said remote memory storage system is transmitted as part of a data payload, wherein said payload comprises normal data, invalidation information and an invalidation header.

5. The system of claim 4, wherein said normal data comprises said specified data annotated with version information.

6. The system of claim 5, wherein said invalidation header is synchronous or asynchronous.

7. The system of claim 5, wherein invalidations having synchronous invalidation headers are processed first.

19

8. The system of claim 5, wherein if said invalidation header is asynchronous, said invalidation is placed in an invalidation queue to be processed.

9. The system of claim 8, wherein a user specifies a frequency for processing said invalidation queue.

10. The system of claim 1, wherein said remote memory storage system is a local disk accessed memory system.

11. The system of claim 1, wherein said remote memory storage system is a network accessed resource.

12. The system of claim 1, wherein said client and said remote memory storage system can switch to a client-server mode for improved efficiency when needed.

13. The system of claim 1, wherein storing pages is tuned.

14. A method for maintaining functional client side data cache coherence in database caches distributed across database servers and clients, comprising:

requesting access to at least one piece of specified data within a data record within a database file utilizing an application resident on a client, and wherein said client is coupled to a local memory cache;

coupling said client to a remote memory storage system, wherein said remote memory storage system is operable to:

store a database file comprising a plurality of data records including one or more linkages between data records;

receive a request for at least one piece of specified data from said client;

retrieve from the database file or a remote cache a data record that includes said at least one piece of specified data; and

transmit said data record to said client along with version information related to data record, the version information comprising a first value representing a version related to the data record and a second value representing a version related to the database file;

receiving the data record and associated version information at the client;

verifying the data record is coherent with any downstream linked data records stored at said client in said local memory cache; and

storing said data record in said local memory cache, if coherency is determined to exist;

20

wherein the database file further comprises a plurality of pages, each page including portions of one or more data records, and wherein the version information is associated with each page individually;

wherein said application uses version information of both the page received and linked pages stored in said local memory cache to verify coherence;

wherein said version information represents a version range and wherein two pages are coherent when said version range of said two pages overlap; and

wherein said version range comprises a version for when a linkage to or from a page was updated and file version for when said a downloaded to the local cache.

15. The method of claim 14, wherein version information is associated with each record individually.

16. The method of claim 14, wherein any writes to said data stored in either said local memory cache or said remote memory storage system are processed by said remote memory storage system prior to executing said write.

17. The method of claim 14, wherein data transmitted from said remote memory storage system comprises a data payload, wherein said payload comprises header information, invalidation meta data, and normal data.

18. The method of claim 17, wherein said normal data comprises said specified data annotated with version information.

19. The method of claim 17, wherein said invalidation meta data comprises a synchronous or asynchronous header.

20. The method of claim 18, wherein invalidations having synchronous invalidation headers are processed first.

21. The method of claim 19, wherein if said invalidation header is asynchronous, said invalidation is placed in an invalidation queue to be processed.

22. The method of claim 21, wherein a user specifies a frequency for processing said invalidation queue.

23. The method of claim 14, wherein said remote memory storage system is a local disk accessed memory system.

24. The method of claim 14, wherein said remote memory storage system is a network accessed resource.

\* \* \* \* \*



57/3,K/25 (Item 19 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013961025 - Drawing available  
WPI ACC NO: 2004-141695/  
Related WPI Acc No: 2003-902187; 2004-033181; 2004-033258; 2004-041839;  
2004-098125; 2004-131046; 2004-374342; 2004-614559  
XRPX Acc No: N2004-113031

**Field replaceable unit identification data conveying method involves receiving separate files including FRU ID data from computer systems, and transmitting received files to remote file server after compression /encryption**

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)

Inventor: GILSTRAP R J; MOCZAR L

**Patent Family** (1 patents, 1 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
US 20030217067	A1	20031120	US 2002381116	P	20020517	200414 B
			US 2002381130	P	20020517	
			US 2002381131	P	20020517	
			US 2002381386	P	20020517	
			US 2002381399	P	20020517	
			US 2002381400	P	20020517	
			US 2003413064	A	20030414	

Priority Applications (no., kind, date): US 2002381400 P 20020517; US 2002381399 P 20020517; US 2002381386 P 20020517; US 2002381131 P 20020517; US 2002381130 P 20020517; US 2002381116 P 20020517; US 2003413064 A 20030414

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20030217067	A1	EN	16	8	Related to Provisional US 2002381116
					Related to Provisional US 2002381130
					Related to Provisional US 2002381131
					Related to Provisional US 2002381386
					Related to Provisional US 2002381399
					Related to Provisional US 2002381400

**Field replaceable unit identification data conveying method involves receiving separate files including FRU ID data from computer systems, and transmitting received files to remote file server after compression /encryption**

**Alerting Abstract** ...NOVELTY - Separate files including field replaceable unit (FRU) identification ( ID ) data are received from the computer systems (102A-102C) by a local file server (104...

...for transmission to a remote file server (108) through a communication medium (106). A FRU ID data transmission notice is then sent to a FRU image repository server (110)...computer system configured to receive FRU ID data files ; carrier medium comprising instructions for conveying FRU ID data ; computer system configured to retrieve and transmit FRU ID data file ; and computer system network...

...USE - For conveying field replaceable unit (FRU) ID data to remote

computer system (claimed) such as high end **server** system used in specialized applications such as distribution of product data to potential consumers through...

...ADVANTAGE - By gathering the FRU **ID data files** in **remote server**, the reliabilities and availabilities of the **server** are improved...

...104 **local server**

...

...108 **remote file server**

...

...110 FRU image **repository server**

...

...112 **FIR database**

**Title Terms...**/Index Terms/Additional Words: **RECEIVE ; ...**

... **ID ; ...**

... **TRANSMIT ; ...**

... **COMPRESS ;**

**Original Publication Data by Authority**

**Original Abstracts:**

...operation (e.g., one or more component operating conditions). One embodiment of the method includes **receiving** a component data file including the component data, and transmitting the component data file (e ...

...data file may be compressed an/or encrypted. For example, a transmitted encrypted compressed component **data file** may be **received** and decrypted to produce a copy of a **compressed component data file**. The **compressed component data file** may be decompressed to produce a copy of the component **data file**, and the component data may be extracted from the component **data file**. Multiple component data files may be received at different times, **compressed** to produce corresponding **compressed component data files**, and the **compressed component data files** may be stored in a designated location. At a designated time, the **compressed component data files** may be retrieved, encrypted to produce corresponding encrypted **compressed component data files**, and the encrypted **compressed component data files** may be **transmitted**. A computer system implementing the method is described. A carrier medium is also described that...

**Claims:**

What is claimed is: b 1 /b . A method for conveying component **data**, comprising: **receiving** a component **data file** comprising the component **data**, wherein the component data identifies the component and is indicative of a state of the component existing during operation of the component; and **transmitting** the component **data file** to a **repository**.





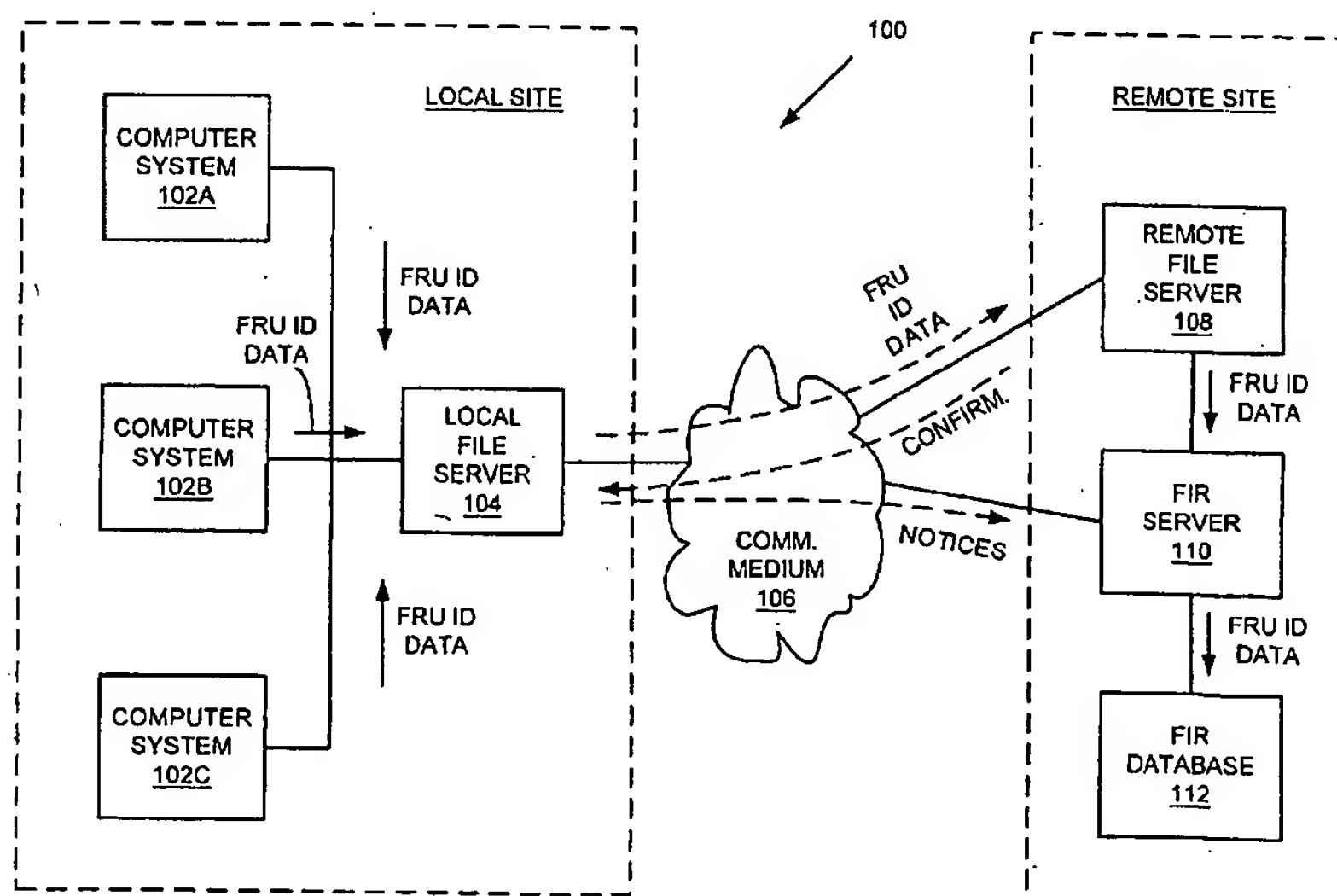
US 20030217067A1

(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0217067 A1**  
Gilstrap et al. (43) **Pub. Date: Nov. 20, 2003**(54) **METHOD AND SYSTEM FOR CONVEYING  
COMPONENT DATA IDENTIFYING A  
COMPONENT AND INDICATING  
COMPONENT OPERATING CONDITIONS****Publication Classification**(51) **Int. Cl.<sup>7</sup>** ..... **G06F 7/00**(52) **U.S. Cl.** ..... **707/100**(75) **Inventors:** **Raymond J. Gilstrap**, Milpitas, CA  
(US); **Lajos Moczar**, Black Forest, CO  
(US)**Correspondence Address:****Lawrence J. Merkel**  
**Meyertons, Hood, Kivlin, Kowert & Goetzel,**  
**P.C.**  
**P.O. Box 398**  
**Austin, TX 78767 (US)**(73) **Assignee:** **Sun Microsystems, Inc.**, Santa Clara,  
CA(21) **Appl. No.:** **10/413,064**(22) **Filed:** **Apr. 14, 2003****Related U.S. Application Data**

(60) Provisional application No. 60/381,399, filed on May 17, 2002. Provisional application No. 60/381,116, filed on May 17, 2002. Provisional application No. 60/381,386, filed on May 17, 2002. Provisional application No. 60/381,131, filed on May 17, 2002. Provisional application No. 60/381,400, filed on May 17, 2002. Provisional application No. 60/381,130, filed on May 17, 2002.

(57) **ABSTRACT**

A method is disclosed for conveying component data identifying a component and indicating one or more states of the component existing during component operation (e.g., one or more component operating conditions). One embodiment of the method includes receiving a component data file including the component data, and transmitting the component data file (e.g., to a component data repository). Prior to the transmitting, the component data file may be compressed and/or encrypted. For example, a transmitted encrypted compressed component data file may be received and decrypted to produce a copy of a compressed component data file. The compressed component data file may be decompressed to produce a copy of the component data file, and the component data may be extracted from the component data file. Multiple component data files may be received at different times, compressed to produce corresponding compressed component data files, and the compressed component data files may be stored in a designated location. At a designated time, the compressed component data files may be retrieved, encrypted to produce corresponding encrypted compressed component data files, and the encrypted compressed component data files may be transmitted. A computer system implementing the method is described. A carrier medium is also described that includes program instructions for carrying out the method. The carrier medium may be, for example, a computer-readable storage medium such as a floppy disk or a compact disk read only memory (CD-ROM) disk.



including the FRU ID data to the local file server 104. The local file server 104 may then perform the method 600 of FIG. 6, thereby transmitting the FRU ID data file to the remote file server 108. The remote file server 108 may then perform the method 700 of FIG. 7, thereby transmitting the FRU ID data to the FIR server 110 (FIG. 1). The FIR server 110 may perform the method 800 of FIG. 8, thereby storing the FRU ID data in the FIR database 112 (FIG. 1).

[0047] The particular embodiments disclosed above are illustrative only, as the invention may be modified and practiced in different but equivalent manners apparent to those skilled in the art having the benefit of the teachings herein. Furthermore, no limitations are intended to the details of construction or design herein shown, other than as described in the claims below. It is therefore evident that the particular embodiments disclosed above may be altered or modified and all such variations are considered within the scope and spirit of the invention. Accordingly, the protection sought herein is as set forth in the claims below.

What is claimed is:

1. A method for conveying component data, comprising:
  - receiving a component data file comprising the component data, wherein the component data identifies the component and is indicative of a state of the component existing during operation of the component; and
  - transmitting the component data file to a repository.
2. The method as recited in claim 1, wherein the receiving comprises:
  - receiving a component data file comprising the component data, wherein a first portion of the component data comprises data that identifies the component, and wherein a second portion of the component data comprises data acquired during operation of the component and associated with an operational state of the component.
3. The method as recited in claim 1, wherein the receiving comprises:
  - receiving a component data file comprising the component data, wherein a first portion of the component data comprises data that identifies the component and is indicative of manufacturing data associated with the component, and wherein a second portion of the component data comprises data acquired during operation of the component and associated with an operational state of the component.
4. The method as recited in claim 1, wherein the receiving comprises:
  - receiving a component data file comprising the component data, wherein a first portion of the component data comprises data that identifies the component, and wherein a second portion of the component data comprises data acquired by the component during operation of the component and associated with an operational state of the component.
5. The method as recited in claim 1, further comprising:
  - receiving the component data file; and
  - extracting the component data from the component data file.
6. A method for conveying component data, comprising:
  - receiving a component data file comprising the component data, wherein the component data identifies the component and is indicative of a state of the component existing during operation of the component;
  - compressing the component data file to produce a compressed component data file, wherein a size of the compressed component data file is less than that of the component data file;
  - encrypting the compressed component data file to produce an encrypted compressed component data file; and
  - transmitting the encrypted compressed component data file.
7. The method as recited in claim 6, wherein the receiving comprises:
  - receiving a component data file comprising the component data, wherein a first portion of the component data comprises data that identifies the component, and wherein a second portion of the component data comprises data acquired during operation of the component and associated with an operational state of the component.
8. The method as recited in claim 6, wherein the receiving comprises:
  - receiving a component data file comprising the component data, wherein a first portion of the component data comprises data that identifies the component and is indicative of manufacturing data associated with the component, and wherein a second portion of the component data comprises data acquired during operation of the component and associated with an operational state of the component.
9. The method as recited in claim 6, wherein the receiving comprises:
  - receiving a component data file comprising the component data, wherein a first portion of the component data comprises data that identifies the component, and wherein a second portion of the component data comprises data acquired by the component during operation of the component and associated with an operational state of the component.
10. The method as recited in claim 6, further comprising:
  - receiving the encrypted compressed component data file;
  - decrypting the encrypted compressed component data file to produce a copy of the compressed component data file;
  - decompressing the compressed component data file to produce a copy of the component data file; and
  - extracting the component data from the component data file.
11. A method for conveying component data, comprising:
  - performing the following for each of a plurality of component data files received at different times and prior to a designated time, wherein each of the component data files corresponds to a different one of a plurality of components:
    - receiving the component data file, wherein the component data file comprises component data that identifies the corresponding component and is indicative

- of a state of the corresponding component existing during operation of the corresponding component;
- compressing the component data file to produce a corresponding compressed component data file, wherein a size of the compressed component data file is less than that of the component data file; and
- storing the compressed component data file in a designated location;
- at the designated time, performing the following for each of the compressed component data files stored in the designated location:
- retrieving the compressed component data file from the designated location;
- encrypting the compressed component data file to produce a corresponding encrypted compressed component data file; and
- transmitting the encrypted compressed component data file.
12. The method as recited in claim 11, further comprising:
- performing the following for each of the encrypted compressed component data files:
- receiving the encrypted compressed component data file;
- decrypting the encrypted compressed component data file to produce a copy of the corresponding compressed component data file;
- decompressing the compressed component data file to produce a copy of the corresponding component data file; and
- extracting the component data from the component data file.
13. A computer system, comprising:
- a memory storing program instructions; and
- a central processing unit (CPU) configured to access the program instructions in the memory and to execute the program instructions;
- wherein when the CPU executes the program instructions, the computer system is configured to receive a component data file comprising component data and to transmit the component data file, wherein the component data identifies a component and is indicative of a state of the component existing during operation of the component.
14. A carrier medium comprising program instructions for conveying component data, wherein the program instructions are operable to implement:
- receiving a component data file comprising the component data, wherein the component data identifies the component and is indicative of a state of the component existing during operation of the component; and
- transmitting the encrypted compressed component data file.
15. The carrier medium as recited in claim 14, wherein the carrier medium is a computer-readable storage medium.
16. The carrier medium as recited in claim 15, wherein the computer-readable storage medium is a floppy disk or a compact disk read only memory (CD-ROM) disk.
17. A computer system, comprising:
- a memory storing program instructions; and
- a central processing unit (CPU) configured to access the program instructions in the memory and to execute the program instructions;
- wherein when the CPU executes the program instructions, the computer system is configured to: (i) receive a component data file comprising component data, wherein the component data identifies a component and is indicative of a state of the component existing during operation of the component, (ii) compress the component data file to produce a compressed component data file, wherein a size of the compressed component data file is less than that of the component data file, (iii) encrypt the compressed component data file to produce an encrypted compressed component data file, and (iv) transmit the encrypted compressed component data file.
18. A carrier medium comprising program instructions for conveying component data, wherein the program instructions are operable to implement:
- receiving a component data file comprising the component data, wherein the component data identifies the component and is indicative of a state of the component existing during operation of the component;
- compressing the component data file to produce a compressed component data file, wherein a size of the compressed component data file is less than that of the component data file;
- encrypting the compressed component data file to produce an encrypted compressed component data file; and
- transmitting the encrypted compressed component data file.
19. The carrier medium as recited in claim 18, wherein the carrier medium is a computer-readable storage medium.
20. The carrier medium as recited in claim 19, wherein the computer-readable storage medium is a floppy disk or a compact disk read only memory (CD-ROM) disk.
21. A carrier medium comprising program instructions for conveying component data, wherein the program instructions are operable to implement:
- performing the following for each of a plurality of component data files received at different times and prior to a designated time, wherein each of the component data files corresponds to a different one of a plurality of components:
- receiving the component data file, wherein the component data file comprises component data that identifies the corresponding component and is indicative of a state of the corresponding component existing during operation of the corresponding component;
- compressing the component data file to produce a corresponding compressed component data file, wherein a size of the compressed component data file is less than that of the component data file; and
- storing the compressed component data file in a designated location;

at the designated time, performing the following for each of the compressed component data files stored in the designated location:

retrieving the compressed component data file from the designated location;

encrypting the compressed component data file to produce a corresponding encrypted compressed component data file; and

transmitting the encrypted compressed component data file.

22. The carrier medium as recited in claim 21, wherein the carrier medium is a computer-readable storage medium.

23. The carrier medium as recited in claim 22, wherein the computer-readable storage medium is a floppy disk or a compact disk read only memory (CD-ROM) disk.

24. A method for conveying component data, comprising:

providing a field replaceable unit having a memory device configured to store component data, wherein the component data identifies the field replaceable unit and is indicative of a state of the field replaceable unit existing during operation of the field replaceable unit;

accessing the field replaceable unit to retrieve the component data;

generating a component data file dependent upon the component data; and

transmitting the component data file.

25. A computer system, comprising:

a field replaceable unit including a memory device configured to store component data, wherein the component data identifies the field replaceable unit and is indicative of a state of the field replaceable unit existing during operation of the field replaceable unit; and

a processing unit operably coupled to the field replaceable unit and to a communication medium, wherein the processing unit is configured to access the memory device, to retrieve the component data from the memory device, to generate a component data file dependent upon the component data, and to transmit the component data file via the communication medium.

26. A system, comprising:

a first computer system coupled to a communication medium and comprising a field replaceable unit, the field replaceable unit having a memory device configured to store component data associated with the field replaceable unit, the first computer system being adapted to access the memory device to retrieve the component data, to generate a component data file dependent upon the component data, and to transmit the component data file via the communication medium; and

a second computer system coupled to the communication medium and configured to receive the component data file via the communication medium, and to extract the component data from the component data file.

27. The system as recited in claim 26, wherein the component data identifies the field replaceable unit and is indicative of a state of the field replaceable unit existing during operation of the field replaceable unit.

28. The system as recited in claim 26, wherein the first computer system is configured to generate and transmit the component data file periodically.

29. The system as recited in claim 26, wherein the second computer system is configured to transmit a component data request to the first computer system via the communication medium, and wherein the first computer system is configured to generate the component data file and to transmit the component data file to the second computer system in response to the component data request.

30. The system as recited in claim 26, wherein the second computer system comprises a component data repository, and wherein the second computer system is configured to store the component data in the component data repository.

31. A method for conveying component data to a remote location, comprising:

receiving a component data file comprising the component data at the remote location, wherein the component data identifies the component and is indicative of a state of the component existing during operation of the component;

extracting the component data from the component data file; and

storing the component data in a repository at the remote location.

32. The method as recited in claim 31, wherein the receiving comprises:

receiving a component data file comprising the component data at the remote location, wherein a first portion of the component data comprises data that identifies the component, and wherein a second portion of the component data comprises data acquired during operation of the component and associated with an operational state of the component.

33. The method as recited in claim 31, wherein the receiving comprises:

receiving a component data file comprising the component data at the remote location, wherein a first portion of the component data comprises data that identifies the component and is indicative of manufacturing data associated with the component, and wherein a second portion of the component data comprises data acquired during operation of the component and associated with an operational state of the component.

34. The method as recited in claim 31, wherein the receiving comprises:

receiving a component data file comprising the component data at the remote location, wherein a first portion of the component data comprises data that identifies the component, and wherein a second portion of the component data comprises data acquired by the component during operation of the component and associated with an operational state of the component.

\* \* \* \* \*



57/3,K/9 (Item 3 from file: 350)  
DIALOG(R) File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0015793726 - Drawing available  
WPI ACC NO: 2006-350096/200636  
Related WPI Acc No: 2005-393604; 2005-675239; 2006-043887; 2006-043914;  
2006-116389; 2006-134290; 2006-153161; 2006-328700; 2006-442310  
XRPX Acc No: N2006-296988

**Mailpiece information sorting method e.g. for letter, involves storing received identification file in lookup table and maintaining service area table database for secondary identification code server**

Patent Assignee: US POSTAL SERVICE (USPO-N)

Inventor: AVANT O L; BRANDT B A; FADELY J D; LITTLE M R

**Patent Family** (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20060096897	A1	20060511	US 1999152194	P	19990831	200636 B
			US 2000652707	A	20000831	
			US 2006326447	A	20060106	

Priority Applications (no., kind, date): US 2000652707 A 20000831; US 1999152194 P 19990831; US 2006326447 A 20060106

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20060096897	A1	EN	48	26	Related to Provisional US 1999152194 Division of application US 2000652707

**Mailpiece information sorting method e.g. for letter, involves storing received identification file in lookup table and maintaining service area table database for secondary identification code server**

#### Original Titles:

Apparatus and methods for processing mailpiece information by an identification code server

**Alerting Abstract ...NOVELTY** - The method involves storing received identification file containing identification ( ID ) tag and file postal code corresponding to mailpiece, in lookup table . A service area table database is maintained with table for secondary ID code server (SICS). A SICS...

... ZIP data file is generated using area table corresponding to SICS to identify ID files in lookup table and transmitted to SICS for updation . ...ADVANTAGE - Automation of mail sorting and other processing tasks are enabled, while reducing cost and delays in mail delivery service...

**Title Terms.../Index Terms/Additional Words:** RECEIVE ; ...

... AREA ; ...

... DATABASE ;

**Class Codes**



International Classification (+ Attributes)  
IPC + Level Value Position Status Version  
... G06F-0007/00  
Manual Codes (EPI/S-X): T01-J10B2A ...

... T01-J17

#### Original Publication Data by Authority

#### Original Abstracts:

...provide for processing mailpiece information in an identification code sorting system by an identification code **server**. In one embodiment, a **primary** identification code **server** **receives** an identification **file** containing identification **information** uniquely corresponding to a mailpiece. In this embodiment, the **primary** identification code **server** processes the mailpiece **information** and may **send** the identification **file** to a **secondary** identification code **server**. In another embodiment, a **secondary** identification code **server** **receives** an identification file from a **primary** identification code **server** and processes the mailpiece information.

#### Claims:

What is claimed is: b 1 /b . A method of processing mailpiece information by a **primary** identification code **server**, comprising the steps of: **receiving** an identification **file** corresponding to a mailpiece from an image control unit, wherein the identification file contains a **file** identification code and a **file postal** code; storing the identification **file** in a **lookup table**; maintaining a service **area table database** with a service area table for a secondary identification code **server**; and updating **the** secondary identification **code** server, wherein **the** updating step further comprises the substeps of: generating a **data file** using the **service** area table corresponding to **the** secondary identification **code** server to identify identification files in **the** **lookup table**; and **transmitting** **the** data file to the secondary identification **code** server.



## CONCLUSION

[0103] As described above, therefore, it will be apparent to those skilled in the art that various modifications and variations can be made in the methods and apparatus of the present invention without departing from the spirit and scope of the invention. Thus, it is intended that the present invention cover the modifications and variations of this invention, provided they come within the scope of the appended claims and their equivalents. In this context, equivalents mean each and every implementation for carrying out the functions recited in the claims, even if not explicitly described herein.

What is claimed is:

1. A method of processing mailpiece information by a primary identification code server, comprising the steps of:

receiving an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;

storing the identification file in a lookup table;

maintaining a service area table database with a service area table for a secondary identification code server; and

updating the secondary identification code server, wherein the updating step further comprises the sub-steps of:

generating a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and

transmitting the data file to the secondary identification code server.

2. The method of claim 1, wherein the storing step further comprises revising an old identification file in the lookup table with a revised identification file.

3. The method of claim 1, wherein the file identification code is an ID tag.

4. The method of claim 1, wherein the file postal code is a POSTNET code.

5. The method of claim 1, wherein the secondary identification code server is a SICS server.

6. The method of claim 1, wherein the data file is a SICS\_ZIP data file.

7. The method of claim 1, wherein the service area table contains a plurality of postal codes corresponding to the secondary identification code server.

8. The method of claim 1, wherein the identification file further includes an image capture time.

9. The method of claim 8, wherein the identification file further includes a plurality of status bits that indicate aspects of the identification file.

10. The method of claim 1, wherein the updating step occurs at a predetermined time interval.

11. The method of claim 10, wherein the predetermined time interval is approximately twenty minutes.

12. The method of claim 1, wherein the updating step occurs when a predetermined number of identification files have been received.

13. The method of claim 12, wherein the predetermined number of identification files is approximately twenty thousand.

14. A system for processing mailpiece information by a primary identification code server, comprising:

an identification file receiving component configured to receive an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;

a storing component configured to store the identification file in a lookup table;

a maintaining component configured to maintain a service area table database with a service area table for a secondary identification code server; and

an updating component configured to update the secondary identification code server, wherein the updating component further comprises:

a generating component configured to generate a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and

a transmitting component configured to transmit the data file to the secondary identification code server.

15. The system of claim 14, wherein the storing component further comprises:

a revising component configured to revise an old identification file in the lookup table with a revised identification file.

16. The system of claim 14, wherein the file identification code is an ID tag.

17. The system of claim 14, wherein the file postal code is a POSTNET code.

18. The system of claim 14, wherein the secondary identification code server is a SICS server.

19. The system of claim 14, wherein the data file is a SICS\_ZIP data file.

20. The system of claim 14, wherein the service area table contains a plurality of postal codes corresponding to the secondary identification code server.

21. The system of claim 14, wherein the identification file further includes an image capture time.

22. The system of claim 14, wherein the identification file further includes a plurality of status bits that indicate aspects of the identification file.

23. The system of claim 14, wherein the updating component operates at a predetermined time interval.

24. The system of claim 23, wherein the predetermined time interval is approximately twenty minutes.

25. The system of claim 14, wherein the updating component operates when a predetermined number of identification files have been received.

26. The system of claim 25, wherein the predetermined number of identification files is approximately twenty thousand.

27. A system for processing mailpiece information by a primary identification code server, comprising:

means for receiving an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;

means for storing the identification file in a lookup table;

means for maintaining a service area table database with a service area table for a secondary identification code server; and

means for updating the secondary identification code server, wherein the updating means further comprises:

means for generating a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and

means for transmitting the data file to the secondary identification code server.

28. A computer usable medium having computer readable code embodied therein for processing mailpiece information by a primary identification code server, the computer readable code comprising:

an identification file receiving module configured to receive an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;

a storing module configured to store the identification file in a lookup table;

a maintaining module configured to maintain a service area table database with a service area table for a secondary identification code server; and

an updating module configured to update the secondary identification code server, wherein the updating module further comprises:

a generating module configured to generate a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and

a transmitting module configured to transmit the data file to the secondary identification code server.

\* \* \* \* \*

(12) **United States Patent**  
Avant et al.

(10) **Patent No.:** US 7,060,925 B1  
(45) **Date of Patent:** Jun. 13, 2006

(54) **APPARATUS AND METHODS FOR PROCESSING MAILPIECE INFORMATION BY AN IDENTIFICATION CODE SERVER**

(75) **Inventors:** Oscar Lee Avant, Silver Spring, MD (US); Bruce A. Brandt, Gainesville, VA (US); Jay David Fadely, Palmetto, FL (US); Michael Ray Little, Fairfax, VA (US)

(73) **Assignee:** United States of America Postal Service, Washington, DC (US)

(\*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 781 days.

(21) **Appl. No.:** 09/652,707

(22) **Filed:** Aug. 31, 2000

**Related U.S. Application Data**

(60) **Provisional application No.** 60/152,194, filed on Aug. 31, 1999.

(51) **Int. Cl.**  
G06F 7/06 (2006.01)  
B07C 5/00 (2006.01)

(52) **U.S. Cl.** ..... 209/584; 700/224; 700/226

(58) **Field of Classification Search** ..... 700/224, 700/226; 209/584, 900  
See application file for complete search history.

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

2,709,001 A	5/1955	Stahl
2,719,629 A	10/1955	Robinson
2,895,588 A	7/1959	Van Marle
3,015,389 A	1/1962	Levy
3,038,607 A	6/1962	Eckert, Jr.
3,215,271 A	11/1965	Cecchini
3,266,626 A	8/1966	Simjian
3,320,593 A	5/1967	Hedrick et al.
3,384,755 A	5/1968	Williamson et al.

(Continued)

**FOREIGN PATENT DOCUMENTS**

EP	0 272 355	6/1988
EP	529966	3/1993
JP	63-101982	5/1988
JP	4160581 A2	6/1992
JP	4-326187	11/1992
JP	6154710 A2	6/1994
NL	8501150 A	11/1986

**OTHER PUBLICATIONS**

Avant et al., "Apparatus and Methods for Reading an Identification Code from a Mailpiece," Pending U.S. Appl. No. 09/652,708, filed Aug. 31, 2000.

Avant et al., "Apparatus and Methods for Identifying and Processing Mail Using an Identification Code," Pending U.S. Appl. No. 09/652,709, filed Aug. 31, 2000.

Avant et al., "Apparatus and Methods for Identifying a Mailpiece Using an Identification Code," Pending U.S. Appl. No. 09/651,323, filed Aug. 31, 2000.

Brandt et al., "Apparatus and Methods for Processing Mailpiece Information in a Mail Processing Device Using Sorter Application Software," Pending U.S. Appl. No. 09/652,275, filed Aug. 31, 2000.

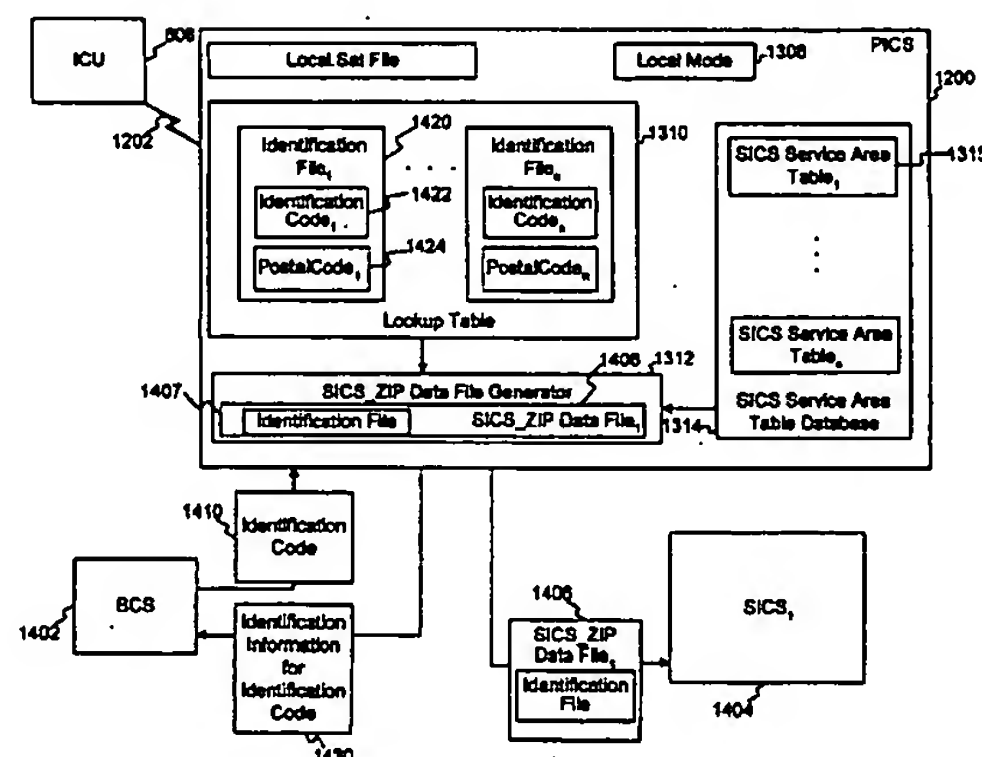
*Primary Examiner*—Joseph Rodriguez

(74) *Attorney, Agent, or Firm*—Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.

(57) **ABSTRACT**

Apparatus and methods consistent with the present invention provide for processing mailpiece information in an identification code sorting system by an identification code server. In one embodiment, a primary identification code server receives an identification file containing identification information uniquely corresponding to a mailpiece. In this embodiment, the primary identification code server processes the mailpiece information and may send the identification file to a secondary identification code server. In another embodiment, a secondary identification code server receives an identification file from a primary identification code server and processes the mailpiece information.

44 Claims, 35 Drawing Sheets





19

directed to a light filter in Light Filter Unit 2310. The first bundle is filtered through a First Light Filter 2312, the second bundle is filtered through a Second Light Filter 2314, and the third bundle is filtered through a Third Light Filter 2316. In this embodiment, the light filters (i.e., First Light Filter 2312, Second Light Filter 2314, and Third Light Filter 2316) respond to different frequencies of the fluorescent spectrum. The analog signals output by Light Filter Unit 2310 are then converted into digital signals by Signal Converter 2318, e.g., an analog/digital converter. Finally, as shown in FIG. 24D, the digital signal from Signal Converter 2318 is passed to Reader Logic Unit 2306, where the digital signal is converted into an ID code corresponding to the ID code on mailpiece 100. Reader Logic Unit 2306 passes the ID code to Port 2320, and the ID code is passed back to BCS 1212.

FIG. 25 shows optional components of an embodiment of a UIDTR such as the UIDTR in FIG. 22. As shown in FIG. 25, an operator 2500 can operate Universal ID Tag Reader 2100 using one or more Light Emitting Diodes 2502 on Reader Head Assembly 2200 and one or more Light Emitting Diodes 2504 and Push Buttons 2506, located on Reader Unit 2202. Light Emitting Diodes 2502 and/or Light Emitting Diodes 2504 can display diagnostic information, such as 'System OK' or 'Power OK,' or function options, such as 'Reset,' to operator 2500. Operator 2500 can use Push Buttons 2506 to display diagnostic information, to select function options or to input other data.

FIG. 26 shows still additional optional components of another embodiment of a UIDTR, such as the UIDTR in FIG. 22. Port 2602 can support, for example, transistor transistor logic (TTL) and Port 2604 can support, for example, differential logic. These optional component ports may enable, for example, UIDTR 2100 to function with an expanded variety of Bar Code Sorters.

## VI. CONCLUSION

As described above, therefore, it will be apparent to those skilled in the art that various modifications and variations can be made in the methods and apparatus of the present invention without departing from the spirit and scope of the invention. Thus, it is intended that the present invention cover the modifications and variations of this invention, provided they come within the scope of the appended claims and their equivalents. In this context, equivalents mean each and every implementation for carrying out the functions recited in the claims, even if not explicitly described herein.

What is claimed is:

1. A method of processing mailpiece information by a primary identification code server, comprising the steps of:
  - receiving an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;
  - storing the identification file in a lookup table;
  - maintaining a service area table database with a service area table for a secondary identification code server;
  - resolving mailpiece information for the mailpiece, wherein the resolving step further comprises the sub-steps of:
    - receiving an identification code from a mail processing device, where the mail processing device obtains the identification code from the mailpiece;
    - processing the identification code to determine the identification information, using the identification file corresponding to the identification code; and

20

- transmitting the identification information to the mail processing device; and
  - updating the secondary identification code server, wherein the updating step further comprises the sub-steps of:
    - generating a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and
    - transmitting the data file to the secondary identification code server.
2. The method of claim 1, wherein the mail processing device is a bar code sorter.
3. The method of claim 1, wherein the storing step further comprises:
  - revising an old identification file in the lookup table with a revised identification file.
4. The method of claim 1, wherein the identification code is an ID tag.
5. The method of claim 1, wherein the file identification code is an ID tag.
6. The method of claim 1, wherein the file postal code is a POSTNET code.
7. The method of claim 1, wherein the service area table contains a plurality of postal codes corresponding to the secondary identification code server.
8. The method of claim 1, wherein the secondary identification code server is a SICS server.
9. The method of claim 1, further comprising the steps of:
  - receiving a delete file message from the mail processing device indicating an identification file to be deleted; and
  - deleting the identification file to be deleted from the lookup table in response to the delete file message.
10. The method of claim 1, wherein the resolving step further comprises the substeps of:
  - connecting to the mail processing device via a telecommunications link;
  - receiving test data from the mail processing device; and
  - confirming the test data.
11. The method of claim 10, wherein the test data consists of nineteen test mailpiece identification codes.
12. The method of claim 1, wherein the identification information transmitting step further comprises the substep of:
  - transmitting the identification file corresponding to the identification code, if the identification code is found in the lookup table.
13. The method of claim 1, wherein the identification information transmitting step further comprises the substep of:
  - transmitting the file postal code corresponding to the identification code, if the identification code is found in the lookup table.
14. The method of claim 1, wherein the identification information transmitting step further comprises the substep of:
  - transmitting an error message to the mail processing device, if the identification code is not found in the lookup table.
15. The method of claim 1, wherein the data file is a SICS\_ZIP data file.
16. The method of claim 1, wherein the identification file further includes an image capture time.
17. The method of claim 16, wherein the identification file further includes a plurality of status bits that indicate aspects of the identification file.

## 21

18. The method of claim 1, wherein the updating step occurs at a predetermined time interval.

19. The method of claim 18, wherein the predetermined time interval is twenty minutes.

20. The method of claim 1, wherein the updating step occurs when a predetermined number of identification files have been received.

21. The method of claim 20, wherein the predetermined number of identification files is twenty thousand.

22. A system for processing mailpiece information by a primary identification code server, comprising:

an identification file receiving component configured to receive an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;

a storing component configured to store the identification file in a lookup table;

a maintaining component configured to maintain a service area table database with a service area table for a secondary identification code server;

a resolving component configured to resolve mailpiece information for the mailpiece, wherein the resolving component further comprises:

an identification code receiving component configured to receive an identification code from a mail processing device, where the mail processing device obtains the identification code from the mailpiece;

a processing component configured to process the identification code to determine the identification information, using the identification file corresponding to the identification code; and

an identification information transmitting component configured to transmit the identification information to the mail processing device; and

an updating component configured to update the secondary identification code server, wherein the updating component further comprises:

a generating component configured to generate a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and

a data file transmitting component configured to transmit the data file to the secondary identification code server.

23. The system of claim 22, wherein the mail processing device is a bar code sorter.

24. The system of claim 22, wherein the storing component further includes:

a revising component configured to revise an old identification file in the lookup table with a revised identification file.

25. The system of claim 22, wherein the identification code is an ID tag.

26. The system of claim 22, wherein the file identification code is an ID tag.

27. The system of claim 22, wherein the file postal code is a POSTNET code.

28. The system of claim 22, wherein the service area table contains a plurality of postal codes corresponding to the secondary identification code server.

29. The system of claim 22, wherein the secondary identification code server is a SICS server.

## 22

30. The system of claim 22, further comprising:

a delete file message receiving component configured to receive a delete file message from the mail processing device indicating an identification file to be deleted; and

a deleting component configured to delete the identification file to be deleted from the lookup table in response to the delete file message.

31. The system of claim 22, wherein the resolving component further includes:

a connecting component configured to connect to the mail processing device via a telecommunications link;

a test data receiving component configured to receive test data from the mail processing device; and

a confirming component configured to confirm the test data.

32. The system of claim 31, wherein the test data consists of nineteen test mailpiece identification codes.

33. The system of claim 22, wherein the identification information transmitting component further comprises:

an identification file transmitting component configured to transmit the identification file corresponding to the identification code, if the identification code is found in the lookup table.

34. The system of claim 22, wherein the identification information transmitting component further comprises:

a file postal code transmitting component configured to transmit the file postal code corresponding to the identification code, if the identification code is found in the lookup table.

35. The system of claim 22, wherein the identification information transmitting component further comprises:

an error message transmitting component configured to transmit an error message to the mail processing device, if the identification code is not found in the lookup table.

36. The system of claim 22, wherein the data file is a SICS\_ZIP data file.

37. The system of claim 22, wherein the identification file further includes an image capture time.

38. The system of claim 37, wherein the identification file further includes a plurality of status bits that indicate aspects of the identification file.

39. The system of claim 22, wherein the updating component operates at a predetermined time interval.

40. The system of claim 39, wherein the predetermined time interval is twenty minutes.

41. The system of claim 22, wherein the updating component operates when a predetermined number of identification files have been received.

42. The system of claim 41, wherein the predetermined number of identification files is twenty thousand.

43. A system for processing mailpiece information by a primary identification code server, comprising:

means for receiving an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;

means for storing the identification file in a lookup table;

means for maintaining a service area table database with a service area table for a secondary identification code server;

means for resolving mailpiece information for the mailpiece, wherein the resolving means further comprises:

means for receiving an identification code from a mail processing device, where the mail processing device obtains the identification code from the mailpiece;

23

means for processing the identification code to determine the identification information, using the identification file corresponding to the identification code; and  
 means for transmitting the identification information to the mail processing device; and  
 means for updating the secondary identification code server, wherein the updating means further comprises:  
 means for generating a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and  
 means for transmitting the data file to the secondary identification code server.

44. A computer usable medium having computer readable code embodied therein for processing mailpiece information by a primary identification code server, the computer readable code comprising:

- an identification file receiving module configured to receive an identification file corresponding to a mailpiece from an image control unit, wherein the identification file contains a file identification code and a file postal code;
- a storing module configured to store the identification file in a lookup table;
- a maintaining module configured to maintain a service area table database with a service area table for a secondary identification code server;

24

a resolving module configured to mailpiece information for the mailpiece, wherein the resolving module further comprises:

- an identification code receiving module configured to receive an identification code from a mail processing device, where the mail processing device obtains the identification code from the mailpiece;
- a processing module configured to process the identification code to determine the identification information, using the identification file corresponding to the identification code; and
- an identification information transmitting module configured to transmit the identification information to the mail processing device; and

an updating module configured to update the secondary identification code server, wherein the updating module further comprises:

- a generating module configured to generate a data file using the service area table corresponding to the secondary identification code server to identify identification files in the lookup table; and
- a data file transmitting module configured to transmit the data file to the secondary identification code server.

\* \* \* \* \*

57/3,K/8 (Item 2 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0015930085 - Drawing available  
WPI ACC NO: 2006-461743/200647  
Related WPI Acc No: 2006-209794; 2006-423141  
XRPX Acc No: N2006-377315

**Computer readable medium for client- server environment, stores data representing data length identifier and tag type, and data of tag type and length described by data length identifier**

Patent Assignee: MICROSOFT CORP (MICT)

Inventor: DHARMARAJAN B

**Patent Family** (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20060129648	A1	20060615	US 2000650104	A	20000829	200647 B
			US 2005313543	A	20051220	

Priority Applications (no., kind, date): US 2000650104 A 20000829; US 2005313543 A 20051220

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20060129648	A1	EN	19	10	Division of application US 2000650104

Division of patent US 7010605

**Computer readable medium for client- server environment, stores data representing data length identifier and tag type, and data of tag type and length described by data length identifier**

**Alerting Abstract ...NOVELTY** - The medium stores **primary data field** containing data representing data length **identifier** and tag type. A **secondary data field** contains data of tag type and length described by **data length identifier**. **USE** - For storing program for encoding session data utilized by **server** and storing session data on client computer...

**...ADVANTAGE** - The amount of **data transferred** between the client and **server** is **minimized**, while maximizing the amount of **information** encoded in the **transferred data**.

#### Original Publication Data by Authority

#### Original Abstracts:

...secret, the length of the secret, the secret itself, and the encoded and encrypted configuration **data**. The session cookie is **transmitted** from a **server** computer to a client computer, where it is stored. Each time the client computer begins a new communications session with the **server** computer that generated the session cookie, the session cookie is **transmitted** from the client computer to the **server** computer. The **server** computer **receives** the session cookie from the client computer and extracts the secret stored in the session cookie. The **server** computer then creates the modified encryption key by inserting the secret into the standard encryption key at the predefined location. The **server** computer

then utilizes the modified encryption key to decrypt the encoded session data stored in the session cookie. Once the encoded session data has been decrypted, the **server** computer decodes the tags contained in the encoded session data. For each tag, the **server** computer determines whether the tag is recognized as a valid tag. If the tag is a valid tag, the **server** computer utilizes the value associated with the tag to configure itself. If the tag is not a valid tag, the **server** computer ignores the tag and attempts to decode the next tag. The **server** computer continues decoding tags until no tags remain to be decoded. A new session cookie may be created and **transmitted** to the client computer. Periodically, the **server** computer may request the new session cookie from the client computer to determine if the communications session between the client computer and the **server** computer is still active. If no response or an invalid session cookie is **received**, the communications session between the client and **server** computers is terminated.

**Claims:**

b 1 /b . A computer-readable medium having stored thereon a data structure, comprising: a **first data field** containing **data** representing a **data length identifier** and a tag type; and a **second data field** containing configuration data of said tag type and having a length described by **said data length identifier**.



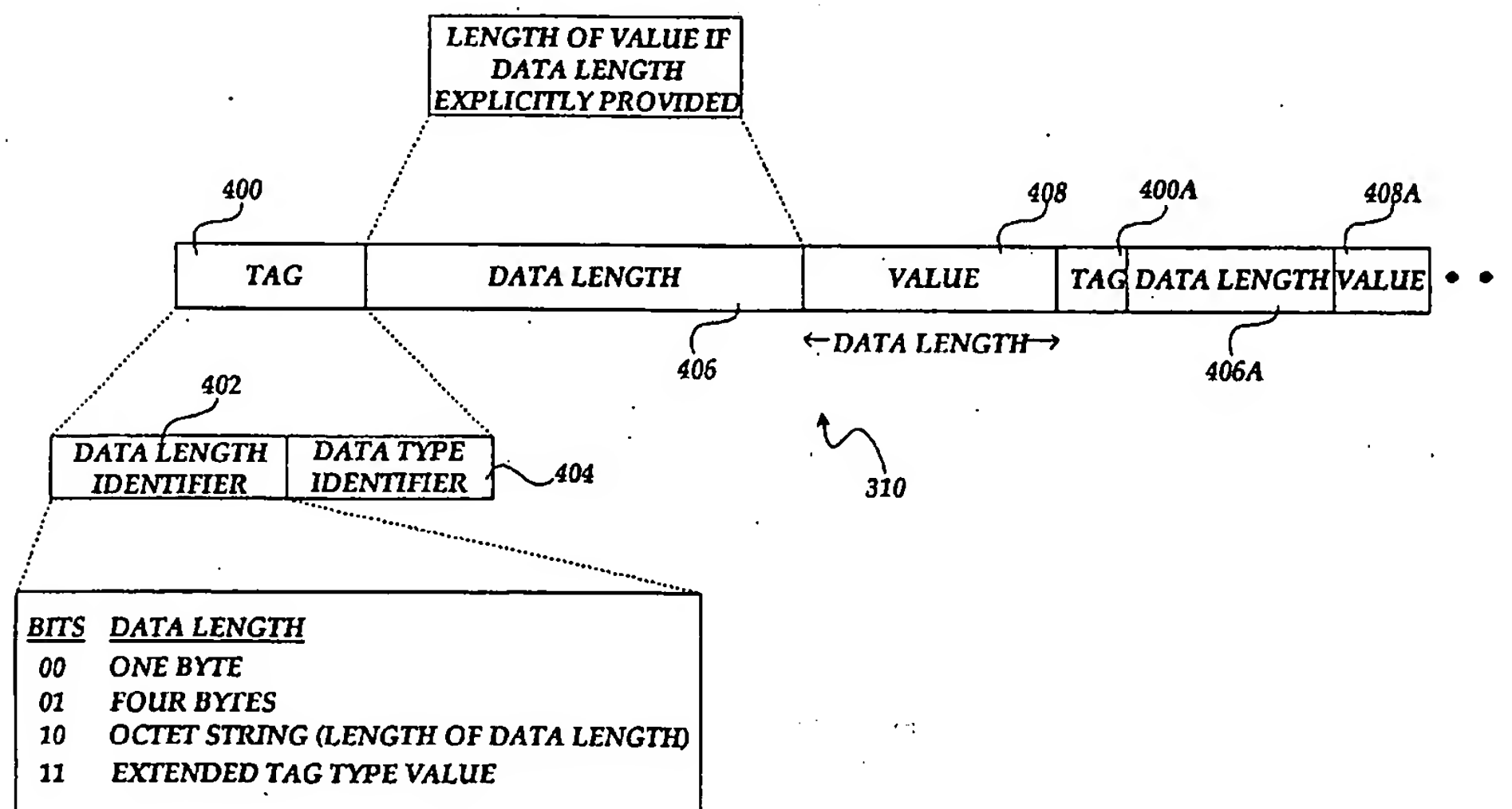


US 20060129648A1

(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2006/0129648 A1**  
(43) **Pub. Date: Jun. 15, 2006****Dharmarajan**(54) **METHOD AND APPARATUS FOR  
ENCODING AND STORING SESSION DATA**(75) **Inventor: Baskaran Dharmarajan, Mountain  
View, CA (US)****Correspondence Address:**  
**CHRISTENSEN, O'CONNOR, JOHNSON,**  
**KINDNESS, PLLC**  
**1420 FIFTH AVENUE**  
**SUITE 2800**  
**SEATTLE, WA 98101-2347 (US)**(73) **Assignee: Microsoft Corporation, Redmond, WA**(21) **Appl. No.: 11/313,543**(22) **Filed: Dec. 20, 2005****Related U.S. Application Data**(62) **Division of application No. 09/650,104, filed on Aug.  
29, 2000, now Pat. No. 7,010,605.****Publication Classification**(51) **Int. Cl.**  
**G06F 15/16 (2006.01)**(52) **U.S. Cl. .... 709/206**(57) **ABSTRACT**

Session data is encoded in a tag-length-value format and encrypted using a modified encryption key. A session cookie

is then formed by concatenating the length of the length of the secret, the length of the secret, the secret itself, and the encoded and encrypted configuration data. The session cookie is transmitted from a server computer to a client computer, where it is stored. Each time the client computer begins a new communications session with the server computer that generated the session cookie, the session cookie is transmitted from the client computer to the server computer. The server computer receives the session cookie from the client computer and extracts the secret stored in the session cookie. The server computer then creates the modified encryption key by inserting the secret into the standard encryption key at the predefined location. The server computer then utilizes the modified encryption key to decrypt the encoded session data stored in the session cookie. Once the encoded session data has been decrypted, the server computer decodes the tags contained in the encoded session data. For each tag, the server computer determines whether the tag is recognized as a valid tag. If the tag is a valid tag, the server computer utilizes the value associated with the tag to configure itself. If the tag is not a valid tag, the server computer ignores the tag and attempts to decode the next tag. The server computer continues decoding tags until no tags remain to be decoded. A new session cookie may be created and transmitted to the client computer. Periodically, the server computer may request the new session cookie from the client computer to determine if the communications session between the client computer and the server computer is still active. If no response or an invalid session cookie is received, the communications session between the client and server computers is terminated.



computer. Routine 1000 then continues from block 1014 to block 1016 where the session timer is reset. The Routine 1000 then continues to block 1004, where the authentication process may begin again.

[0068] If, at block 1012, the Web server computer determines that the session data encoded in the session cookie is not valid, the Routine 1000 continues to block 1018, where the communications session between the Web server computer and the client computer is ended. From block 1018, the Routine 1000 continues to block 1020, where it returns to block 922, shown in FIG. 9. In this manner, the session cookie may be utilized to periodically validate the communications session between the Web server computer and the client computer.

[0069] While illustrative embodiments have been illustrated and described, it will be appreciated that various changes can be made therein without departing from the spirit and scope of the invention.

1. A computer-readable medium having stored thereon a data structure, comprising:

a first data field containing data representing a data length identifier and a tag type; and

a second data field containing configuration data of said tag type and having a length described by said data length identifier.

2. The computer-readable medium of claim 1, wherein said data structure further comprises a plurality of additional data structures comprising one of said first data field and one of said second data field for a plurality of tags.

3. The computer-readable medium of claim 2, wherein said data length identifier comprises the first two bits of said first data field.

4. The computer-readable medium of claim 2, wherein said data length identifier comprises data indicating that the length of said second data field is one byte.

5. The computer-readable medium of claim 2, wherein said data length identifier comprises data indicating that the length of said second data field is four bytes.

6. The computer-readable medium of claim 2, wherein said data length identifier comprises data indicating that said tag type comprises an extended tag type.

\* \* \* \* \*



US007010605B1

(12) **United States Patent**  
**Dharmarajan**

(10) **Patent No.: US 7,010,605 B1**  
(45) **Date of Patent: Mar. 7, 2006**

(54) **METHOD AND APPARATUS FOR  
ENCODING AND STORING SESSION DATA**

(56) **References Cited**

**U.S. PATENT DOCUMENTS**

(75) **Inventor:** Baskaran Dharmarajan, Mountain View, CA (US)  
(73) **Assignee:** Microsoft Corporation, Redmond, WA (US)

5,367,573 A \* 11/1994 Quimby ..... 713/167  
5,623,546 A \* 4/1997 Hardy et al. .... 713/193  
5,689,565 A \* 11/1997 Spies et al. .... 713/189  
6,041,357 A \* 3/2000 Kunzelman et al. .... 709/228  
6,374,359 B1 \* 4/2002 Shrader et al. .... 713/201  
6,557,038 B1 \* 4/2003 Becker et al. .... 709/227

(\*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 852 days.

\* cited by examiner

*Primary Examiner*—Ayaz Sheikh

*Assistant Examiner*—Brandon Hoffman

(74) *Attorney, Agent, or Firm*—Christensen O'Connor Johnson Kindness PLLC

(21) **Appl. No.:** 09/650,104

(22) **Filed:** Aug. 29, 2000

(57) **ABSTRACT**

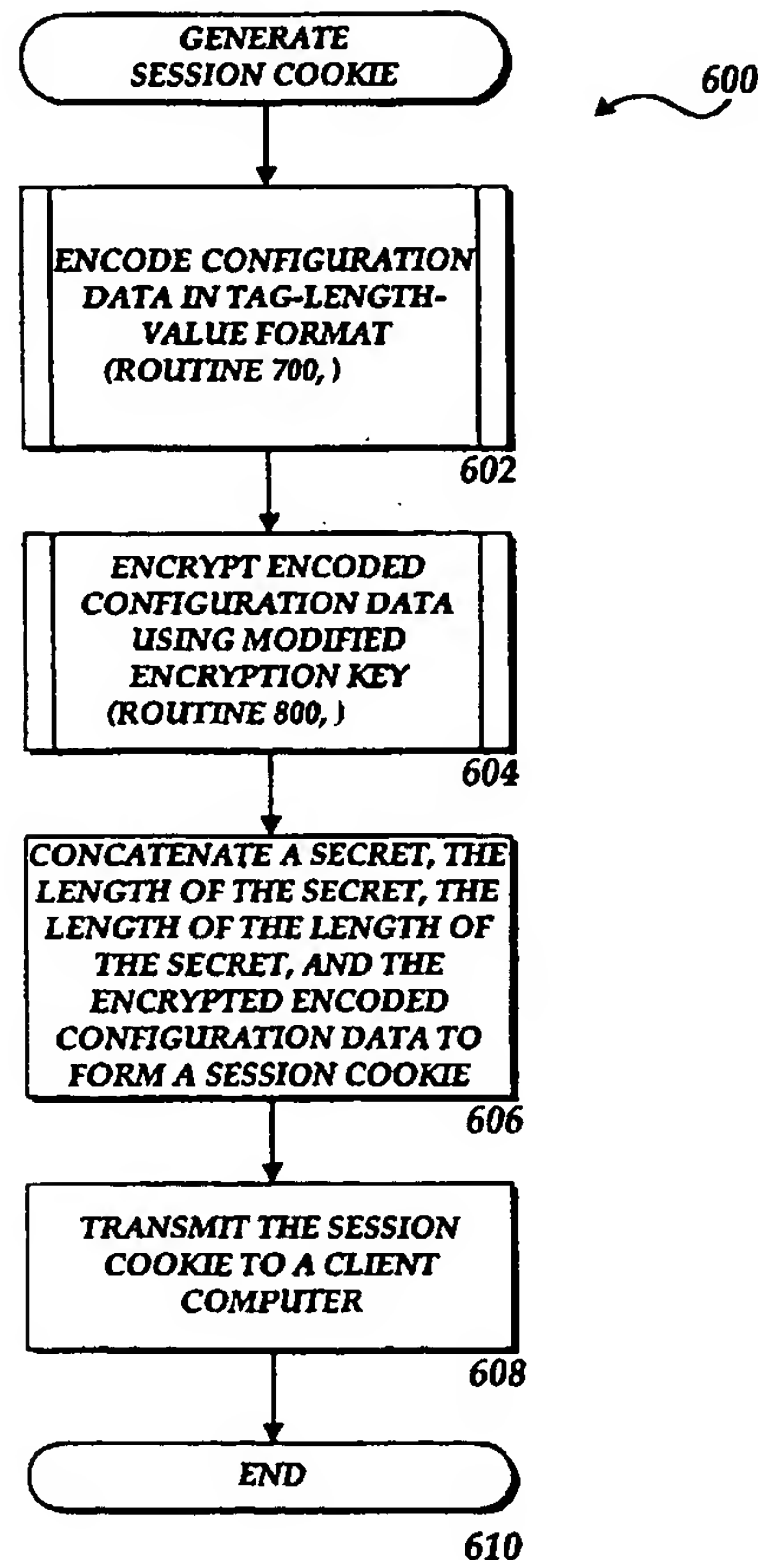
(51) **Int. Cl.**  
**G06F 15/16** (2006.01)  
**H04L 9/00** (2006.01)

Session data is encoded in a tag-length-value format and encrypted using a modified encryption key. A session cookie is then formed by concatenating the length of the length of the secret, the length of the secret, the secret itself, and the encoded and encrypted configuration data. The session cookie is transmitted from a server computer to a client computer, where it is stored.

(52) **U.S. Cl.** ..... 709/227; 713/162  
(58) **Field of Classification Search** ..... 713/162;  
345/745; 709/227

See application file for complete search history.

**13 Claims, 10 Drawing Sheets**



13

Routine 1000 for authenticating a session cookie is described below with reference to FIG. 10. From block 920, the Routine 900 continues to block 922, where it ends.

Referring now to FIG. 10, an illustrative Routine 1000 for authenticating a session cookie will be described. Routine 1000 begins at block 1002, where a session timer is started at the Web server computer. The session timer utilizes a real time clock to determine the amount of time that has elapsed since the session timer was started. The session timer may also be set to elapse after a predetermined amount of time. From block 1002, the Routine 1000 continues to block 1004, where the Web server computer determines if the session timer has elapsed. If the session timer has not elapsed, the Routine 1000 branches back to block 1004, where another determination is made. If the session timer has elapsed, the Routine 1000 continues to block 1006, where the Web server computer requests the session cookie from the client computer.

From block 1006, the Routine 1000 continues to block 1008, where the Web server computer determines whether the client computer has responded to the request for the session cookie or whether the request has timed-out. If the request has timed-out, the Routine 1000 branches to block 1018. If the request has not timed-out, the Routine 1000 continues to block 1010. At block 1010, the Web server computer decrypts the session cookie and decodes the session data. From block 1010, the Routine 1000 continues to block 1012, where the Web server computer determines whether the session data encoded within the session cookie is valid. If the session data encoded within the session cookie is valid, the Routine 1000 branches to block 1014, where the Web server computer generates a new session cookie and transmits the new session cookie to the client computer. Routine 1000 then continues from block 1014 to block 1016 where the session timer is reset. The Routine 1000 then continues to block 1004, where the authentication process may begin again.

If, at block 1012, the Web server computer determines that the session data encoded in the session cookie is not valid, the Routine 1000 continues to block 1018, where the communications session between the Web server computer and the client computer is ended. From block 1018, the Routine 1000 continues to block 1020, where it returns to block 922, shown in FIG. 9. In this manner, the session cookie may be utilized to periodically validate the communications session between the Web server computer and the client computer.

In light of the above, it should be appreciated by those skilled in the art that the present invention provides a method, system, apparatus, and computer-readable medium for encoding and storing session data for a server computer. While an actual embodiment of the invention has been illustrated and described, it will be appreciated that various changes can be made therein without departing from the spirit and scope of the invention.

The invention claimed is:

1. A method for storing session data on a client computer, comprising:
  - encoding said session data in a tag-length-value format to create encoded configuration data, said encoded configuration data including a time stamp;
  - encrypting said encoded configuration data using a modified encryption key to create encrypted encoded configuration data;
  - concatenating a secret, a value that represents the length of the secret, and a value that represents the length of the length of the secret with said encrypted encoded configuration data to form a session cookie;
  - transmitting said session cookie to said client computer;

14

requesting said session cookie from said client computer; receiving said session cookie from said client computer; extracting said secret from said session cookie; creating said modified encryption key by inserting said secret extracted from said session cookie into a standard encryption key at a predefined location; and decrypting said session data from said cookie using said modified encryption key.

2. The method of claim 1, wherein said modified encryption key comprises a standard encryption key with said secret inserted at a predefined location.

3. The method of claim 2, further comprising:

decoding a tag from said session data; determining whether said tag comprises a valid tag; and in response to determining that said tag comprises a valid tag, configuring said server using data contained in said tag.

4. The method of claim 3, further comprising:

in response to determining that said tag does not comprise a valid tag, determining whether additional tags remain to be decoded from said encoded configuration data; and

in response to determining that additional tags remain to be decoded, decoding a next tag and determining whether said next tag comprises a valid tag.

5. The method of claim 4, further comprising:

in response to determining that said next tag comprises a valid tag, configuring said server using data contained in said next tag.

6. The method of claim 5, further comprising:

in response to determining that additional tags do not remain to be decoded, periodically authenticating said session cookie.

7. The method of claim 6, wherein periodically authenticating said session cookie comprises:

starting a session timer;

determining whether said session timer has elapsed; and in response to determining that said session timer has elapsed,

(i) requesting said session cookie from said client computer,

(ii) decrypting and decoding a tag contained in said session cookie, and

(iii) determining whether said tag comprises a valid tag.

8. The method of claim 7, further comprising:

in response to determining that said tag comprises a valid tag,

(i) generating a new session cookie,

(ii) transmitting said new session cookie to said client computer, and

(iii) resetting said session timer.

9. The method of claim 7, further comprising:

in response to determining that said tag does not comprise a valid tag, ending a communications session between said server computer and said client computer.

10. A computer-readable medium containing computer-readable instructions which, when executed by a computer, perform the method of claim 1.

11. A computer-readable medium containing computer-readable instructions which, when executed by a computer, perform the method of claim 2.

12. A computer-controlled apparatus for performing the method of claim 1.

13. A computer-controlled apparatus for performing the method of claim 2.

\* \* \* \* \*

57/3,K/6 (Item 6 from file: 347)  
DIALOG(R)File 347:JAPIO  
(c) 2006 JPO & JAPIO. All rts. reserv.

05904414 \*\*Image available\*\*  
METHOD AND DEVICE FOR REFLECTING DATA WEAR HOUSE WITH DATA

PUB. NO.: 10-187514 [JP 10187514 A]  
PUBLISHED: July 21, 1998 (19980721)  
INVENTOR(s): MATSUMURA EIICHI  
APPLICANT(s): HITACHI LTD [000510] (A Japanese Company or Corporation), JP  
(Japan)  
APPL. NO.: 08-344997 [JP 96344997]  
FILED: December 25, 1996 (19961225)

#### ABSTRACT

PROBLEM TO BE SOLVED: To provide consistency between the contents of extraction source data base and the contents of reflection destination data base while matching it with the purpose of user without applying line load as much as...

... reflection is registered concerning a certain key item on a certain table in a reflection destination data base 142. Then, data update information 121 of extraction source system 112 is acquired from an existent data update information extracting device 120 and while using this information and the information of relation between the extraction source data base 113 and a reflection destination data base 142 provided from meta data 143, the differential value of reflection destination data base 142 from the time of last reflection with object data is found and it is...



# METHOD AND DEVICE FOR REFLECTING DATA WEAR HOUSE WITH DATA

Publication number: JP10187514

Publication date: 1998-07-21

Inventor: MATSUMURA EIICHI

Applicant: HITACHI LTD

Classification:

- international: G06F12/00; G06F17/30; G06F12/00; G06F17/30;  
(IPC1-7): G06F12/00; G06F12/00; G06F17/30

- european:

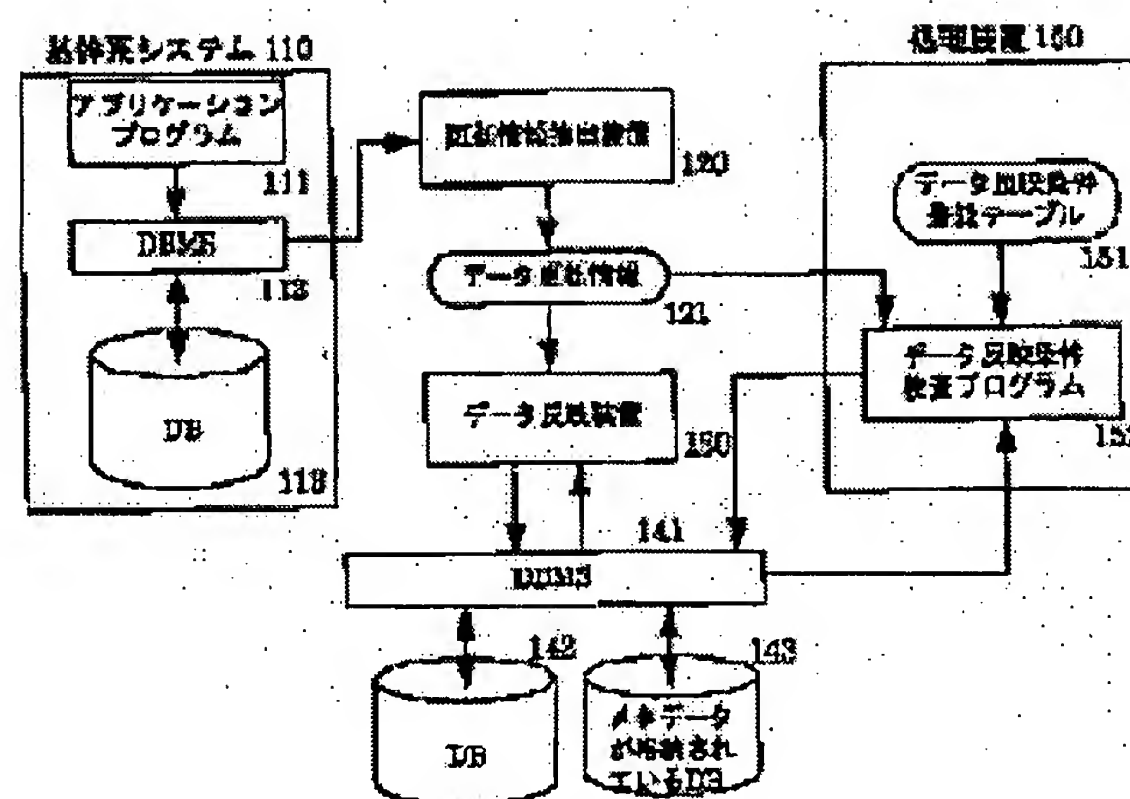
Application number: JP19960344997 19961225

Priority number(s): JP19960344997 19961225

Report a data error here

## Abstract of JP10187514

**PROBLEM TO BE SOLVED:** To provide consistency between the contents of extraction source data base and the contents of reflection destination data base while matching it with the purpose of user without applying line load as much as possible. **SOLUTION:** Several data reflection conditions are previously registered on a data reflection condition registration table 151 of processor 150. For example, the differential value of certain column from the time of last reflection is registered concerning a certain key item on a certain table in a reflection destination data base 142. Then, data update information 121 of extraction source system 112 is acquired from an existent data update information extracting device 120 and while using this information and the information of relation between the extraction source data base 113 and a reflection destination data base 142 provided from meta data 143, the differential value of reflection destination data base 142 from the time of last reflection with object data is found and it is inspected whether that differential value exceeds a registered value or not. When it exceeds the registered value, the reflection destination system 141 is reflected with the differential value.



Data supplied from the esp@cenet database - Worldwide

57/3,K/156 (Item 150 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c). 2006 The Thomson Corporation. All rts. reserv.

0008178107 - Drawing available  
WPI ACC NO: 1997-280581/199725  
XRPX Acc No: N1997-232523

Compressed and uncompressed data transfer method between storage system - involves compressing data record in data stream at first storage system without assistance from host system or second storage system, and updating meta - data for data record that has been compressed to indicate that data is compressed

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: CARREIRO P P; FISH R R; NOWLEN D R

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 5630092	A	19970513	US 1994326407	A	19941020	199725 B
			US 1996581719	A	19960102	

Priority Applications (no., kind, date): US 1994326407 A 19941020; US 1996581719 A 19960102

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 5630092	A	EN	16	8	Continuation of application US 1994326407

Compressed and uncompressed data transfer method between storage system...

...involves compressing data record in data stream at first storage system without assistance from host system or second storage system, and updating meta - data for data record that has been compressed to indicate that data is compressed

#### Original Titles:

System and method for transferring compressed and uncompressed data between storage systems.

**Alerting Abstract** ...The method involves storing in a first storage system meta - data about each data record including whether a data record is compressed. Data record in the data stream is compressing at the first storage system without assistance from the host system or the second storage system. The meta-data for a data record that has been compressed is updated to indicate that the data record is compressed. The data stream is sent with the meta - data from the first storage system to the second storage system...

...The meta - data and data stream are stored on a storage device at the second storage system. The meta - data at the second storage system reads that at least one data record in the data stream has been compressed. The length of the data record when compressed and the length of the data record when non-compressed are stored as part of the meta -data. Both compressed and non-compressed data records are transmitted as part of the data stream...

...ADVANTAGE - Improves performance when **transferring data** from primary to secondary storage by enabling **data** to be **transmitted in compressed** form.

Title Terms/Index Terms/Additional Words: **COMPRESS ; ...**

... **TRANSFER ; ...**

... **UPDATE ; ...**

... **META ;**

Original Publication Data by Authority

**Original Abstracts:**

A system and method are provided for sharing a data stream between a **first** data storage system in communication with a second **data storage** system. **Meta - data** is associated with each record of the data stream. Zero or more records of the **data** stream are **compressed** at the first storage system. The **meta -data** for the compressed data record is updated to indicate that the data record has been compressed. The data stream including the compressed **data record** with the **updated meta - data** is **sent** to the second storage system. The **meta - data** and the **data** stream records are stored on a storage device at the **second data storage** system.

**Claims:**

b Claim 7. /b A system for sharing a data stream of **data records** between a **first** and **second** storage system in communication with each other and each storage system wherein both **compressed** and non- **compressed data records** are **transmitted** as part of the **data** stream in communication with at least one host system, each storage system including at least one storage device having a plurality of **data records** stored on the storage device, said system comprising: means for associating with at least one **data record** stored on the **first** storage system a record length **indicator ; compression** means for **compressing** and decompressing at least one **data record** at the **first** storage system independent of the host system and the second storage system; means for associating with a **compressed** data record stored on the first storage system a compressed **data record length indicator ;** means for sending the **data** stream including the compressed **data record** with the record length **indicator** and a **compressed** record length **indicator** from the first storage system to the **second** storage system; means for storing the **compressed data record** and **indicators** on a storage device at the **second** storage subsystem; means for retrieving a **data record** from a storage device at the **second** storage system; means for identifying a **compressed data record** , at the **second** storage system independent of the first storage system and the host system; means for decompressing **the compressed data** record at **the** second storage system independent of the first storage system and the host system; and **means** for **sending** a data stream to a requesting host system directly from the second storage system independent of the first storage system and the host system; **means** for **transferring** the data stream from the second storage system to a third storage system wherein **all** data records in the data stream are in a non-compressed state, **means** for **transferring** a data record from the second storage system to the third storage subsystem in a compressed state wherein the third storage subsystem is unaware of the compressed state and **stores** **the** data record on a storage device controlled by the third storage system

independent of the...

57/3,K/138 (Item 132 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0009207351 - Drawing available  
WPI ACC NO: 1999-132510/199911  
XRPX Acc No: N1999-096482

**Push based method for delivering information from Internet to client - involves client subscribing to channels and configuring information required and service provider delivering information via coaches as needed**

Patent Assignee: KEMPER D J (KEMP-I); LAMBERT M L (LAMB-I); TIBCO INC (TIBC-N); TIBCO SOFTWARE INC (TIBC-N); VAN DER RIJN D J G (VRIJ-I); VERKLER J L (VERK-I)

Inventor: KEMPER D J; LAMBERT M L; VAN DER RIJN D J G; VERKLER J L

**Patent Family** (6 patents, 80 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
WO 1999004345	A1	19990128	WO 1998US15131	A	19980721	199911 B
AU 199885788	A	19990210	AU 199885788	A	19980721	199925 E
US 6038601	A	20000314	US 1997897786	A	19970721	200020 E
EP 996893	A1	20000503	EP 1998936965	A	19980721	200026 E
			WO 1998US15131	A	19980721	
US 20030149737	A1	20030807	US 1997897786	A	19970721	200358 E
			US 1999379376	A	19990823	
			US 2002299614	A	20021118	
US 6629138	B1	20030930	US 1997897786	A	19970721	200367 E
			US 1999379376	A	19990823	

Priority Applications (no., kind, date): US 2002299614 A 20021118; US 1999379376 A 19990823; US 1997897786 A 19970721

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing	Notes
--------	------	-----	----	-----	--------	-------

WO 1999004345	A1	EN	91	8		
---------------	----	----	----	---	--	--

National Designated States,Original: AL AM AT AZ BA BB BG BR BY CA CH CN  
CU CZ DE DK EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK  
LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ  
TM TR TT UA UG UZ VN YU ZW

Regional Designated States,Original: AT BE CH CY DE DK EA ES FI FR GB GH  
GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW

AU 199885788	A	EN			Based on OPI patent	WO 1999004345
--------------	---	----	--	--	---------------------	---------------

EP 996893	A1	EN			PCT Application	WO 1998US15131
-----------	----	----	--	--	-----------------	----------------

Based on OPI patent WO 1999004345

Regional Designated States,Original: BE DE ES FR GB NL SE

US 20030149737	A1	EN			Continuation of application	US 1997897786
----------------	----	----	--	--	-----------------------------	---------------

Division of application US 1999379376

US 6629138	B1	EN				
------------	----	----	--	--	--	--

Continuation of patent US 6038601

Continuation of application US

Continuation of patent US 6038601

**Alerting Abstract** ...the information required. Service providers feed the information via caching systems that are managed by **meta data** to deliver the required **information** .

...



...OF DRAWING(S) - DESCRIPTION OF DRAWING(S) - System overview (100) Client web browser; (204) Caching **server** .

#### Original Publication Data by Authority

#### Original Abstracts:

...one embodiment of the present invention, a method and apparatus for maintaining statistics on a **server** (204) is disclosed. According to an alternative embodiment, a method and apparatus (204) is disclosed for predicting data that a client device (100) may request from a **server** on a network. In **another** embodiment of the present invention, a method and apparatus (204) is disclosed for managing bandwidth...

...one embodiment of the present invention, a method and apparatus for maintaining statistics on a **server** is disclosed. According to an alternative embodiment, a method and apparatus is disclosed for predicting data that a client device may request from a **server** on a network. In **another** embodiment of the present invention, a method and apparatus is disclosed for managing bandwidth between...

...A method for maintaining statistics on a **remote server** . The method includes **receiving** statistics from client devices that are coupled to the **remote server** . The statistics are associated with **data** on the **remote server** . The **remote server** **receives** statistics either when a user accesses the **data** on the **remote server** or when collected statistics associated with the **data** previously **downloaded** into a cache on each client device is **uploaded** to the **remote server** from each client device. The method also includes **updating** the statistics on the **remote server** in response to either a user access of the data on the **remote server** or a receipt of the collected statistics from each client device; and **downloading** the **updated** statistics to each client device...

...one embodiment of the present invention, a method and apparatus for maintaining statistics on a **server** (204) is disclosed. According to an alternative embodiment, a method and apparatus (204) is disclosed for predicting data that a client device (100) may request from a **server** on a network. In **another** embodiment of the present invention, a method and apparatus (204) is disclosed for managing bandwidth...

#### Claims:

We claim: b 1 /b . A method for validating a collection of **data** , the method including: **receiving** a request for **data** in the collection of data, each data in the collection of data associated to an...

...a table of contents (TOC);examining the TOC to determine whether the TOC is expired; **updating** the TOC if the TOC is expired; andvalidating the collection of data with the...

...5. A method for improving user perceived response time when a client device requests data **from** a server on a network, the method **including** :analyzing initial data requested **from** the server by one or more application programs running on the client device in order to identify references to various data **on** the server;assigning a weight to each said identified reference which represents a likelihood relative...

...identified by each said identified reference to the client device in anticipation of an actual **request** for the data by an **application**

program, said downloading being performed for each said identified  
reference in order of decreasing weight, wherein the likelihood that said  
application program will request said various...

57/3,K/141 (Item 135 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0009026933 - Drawing available  
WPI ACC NO: 1998-583871/199849  
XRPX Acc No: N1998-454827

**DASD file system copy apparatus** - determines source location on data storage subsystem, target location on data storage subsystem and identifies extents of both with data transmitted to data storage subsystem representative of assignments of DASD full tracks

Patent Assignee: STORAGE TECHNOLOGY CORP (STOS)

Inventor: TOMSULA P J; WHITE M W

**Patent Family** (5 patents, 19 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 1998048348	A1	19981029	WO 1998US7457	A	19980415	199849	B
EP 974095	A1	20000126	EP 1998915553	A	19980415	200010	E
			WO 1998US7457	A	19980415		
US 6108749	A	20000822	US 1997844046	A	19970418	200042	E
EP 974095	B1	20020220	EP 1998915553	A	19980415	200214	E
			WO 1998US7457	A	19980415		
DE 69803923	E	20020328	DE 69803923	A	19980415	200229	E
			EP 1998915553	A	19980415		
			WO 1998US7457	A	19980415		

Priority Applications (no., kind, date): US 1997844046 A 19970418

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 1998048348	A1	EN	18	2	
National Designated States,Original: JP					
Regional Designated States,Original: AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE					
EP 974095	A1	EN			PCT Application WO 1998US7457 Based on OPI patent WO 1998048348
Regional Designated States,Original: DE FR					
EP 974095	B1	EN			PCT Application WO 1998US7457 Based on OPI patent WO 1998048348
Regional Designated States,Original: DE FR					
DE 69803923	E	DE			Application EP 1998915553 PCT Application WO 1998US7457 Based on OPI patent EP 974095 Based on OPI patent WO 1998048348

**Alerting Abstract** ...selected data file, and the apparatus has memory space for the copy of the selected data file .

...

...A data processor, responsive to the data storage subsystem copies the selected data file , and updates meta data associated with the data file and known by the data processor. The memory allocator determines a location and an extent of the selected data file . Control messages are transmitted to the data storage subsystem to initiate the copy of the selected data file .

...

...ADVANTAGE - Enables data processor to manage **data file** copy function of disk **data storage** subsystem in manner which **minimises** expenditure of **data** processor resources.

#### Class Codes

(Additional/Secondary): G06F-013/00

Manual Codes (EPI/S-X): T01-G03 ...

#### Original Publication Data by Authority

#### Original Abstracts:

The DASD file system copy system functions to enable the data processor to manage the **data file** copy function of a disk **data storage** subsystem in a manner that **minimizes** the expenditure of **data** processor resources. This is accomplished by the DASD file system copy system determining the **source** location on the **data storage** subsystem, the **target** location on the **data storage** subsystem and identifying the extents of both. The DASD file system copy system then **transmits** data to the **data storage** subsystem, representative of the assignment of DASD full tracks from the **source** location on the **data storage** subsystem as well as DASD full tracks from the **target** location on the **data storage** subsystem. The data processor based DASD file system copy system then uses ECAM channel programs to instruct the **data storage** subsystem to perform the **data file** copy operation using snapshot track **pointer** copy operations. Upon conclusion of the **data file** copy operation by the **data storage** subsystem, the DASD file system copy system **updates** the **meta data** required to complete the **data file** copy operation...

...The DASD file system copy system functions to enable the data processor to manage the **data file** copy function of a disk **data storage** subsystem in a manner that **minimizes** the expenditure of **data** processor resources. This is accomplished by the DASD file system copy system determining the **source** location on the **data storage** subsystem, the **target** location on the **data storage** subsystem and identifying the extents of both. The DASD file system copy system then **transmits** data to the **data storage** subsystem, representative of the assignment of DASD full tracks from the **source** location on the **data storage** subsystem as well as DASD full tracks from the **target** location on the **data storage** subsystem. The data processor based DASD file system copy system then uses ECAM channel programs to instruct the **data storage** subsystem to perform the **data file** copy operation using snapshot track **pointer** copy operations. Upon conclusion of the **data file** copy operation by the **data storage** subsystem, the DASD file system copy system **updates** the **meta data** required to complete the **data file** copy operation...

...The DASD file system copy system functions to enable the data processor to manage the **data file** copy function of a disk **data storage** subsystem in a manner that **minimizes** the expenditure of **data** processor resources. This is accomplished by the DASD file system copy system determining the **source** location on the **data storage** subsystem, the **target** location on the **data storage** subsystem and identifying the extents of both. The DASD file system copy system then **transmits** data to the **data storage** subsystem, representative of the assignment of DASD full tracks from the **source** location on the **data storage** subsystem as well as DASD full tracks from the **target** location on the **data storage** subsystem. The data processor based DASD file system copy system then uses ECAM channel programs to instruct the **data storage** subsystem to perform

the **data file** copy operation using snapshot track **pointer** copy operations. Upon conclusion of the **data file** copy operation by the **data storage subsystem**, the **DASD file system copy system** **updates** the **meta data** required to complete the **data file** copy operation.

**Claims:**

...sowie einer Kopie der ausgewählten Datendatei zuweist; eine Einrichtung, die Steuermeldungen an das Datenspeicherungs-Teilsystem **sendet**, um die ausgewählte Datendatei und Speicherplatz für die Kopie der ausgewählten Datendatei zu identifizieren; und...

... **A data file system copy apparatus** for the copying of **data files** stored on a **dynamically mapped virtual memory data storage subsystem** having a rewriteable memory space, **which data file copy apparatus** is extant on a **data processor** connected to the **data storage subsystem** which is operational to instantaneously create a copy of a **selected data file** independent of said data processor, **said data file copy apparatus** comprising: means for allocating memory in said rewriteable memory space for a **selected data file** written thereon as well as for a copy of said **selected data file**; means for transmitting control messages to **said data storage subsystem** to identify said **selected data file**, and memory space for said copy of said **selected data file**; and means, responsive to said data storage subsystem copying **said selected data file**, for **updating** meta data associated with **said data file** and known by said data processor...

...copie des fichiers de données mémorisés sur un sous-système de mémorisation de données à **mémoire** virtuelle mappée dynamiquement comportant un espace de mémoire reinscriptible, lequel dispositif de copie de fichiers...

...moyen destiné à transmettre des messages de commande audit sous-système de mémorisation de données **afin** d'identifier ledit fichier de données sélectionné, et un espace de mémoire pour ladite copie...

...ledit sous-système de mémorisation de données dudit fichier de données sélectionné, afin de mettre à jour des meta données associées audit fichier de données et connues dudit processeur de données...

... **Claim 6 . /b A data file system copy apparatus** for the copying of data files stored on a **dynamically mapped virtual memory data storage subsystem** having a rewriteable **memory space**, **which data file copy apparatus** is **extant** on a **data processor** **connected** to the **data storage subsystem** which is operational to instantaneously create a copy of a **selected data file** independent of said **data processor**, **said data file copy apparatus** comprising: means for allocating memory in said rewriteable memory space **for a selected data file** written thereon as well as for a copy of **said selected data file**; means for transmitting control **messages** to said data storage subsystem to **identify said selected data file**, and memory space for said copy of **said selected data file**; and **means**, responsive to said data storage subsystem copying **said selected data file**, for updating meta data associated with said data file **and** known by said data **processor**, comprising: means for updating **at least one of data file meta data** comprising: Virtual Table of Contents (VTOC); Volume labels; **VTOC** index; Virtual Volume Data Set (VVDS) data.



57/3,K/118 (Item 112 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0009967980 - Drawing available  
WPI ACC NO: 2000-270271/  
Related WPI Acc No: 1999-633516  
XRPX Acc No: N2000-202380

**Processing method for requests to access a data storage subsystem that maintains managed files in a storage management system**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: CANNON D M

**Patent Family** (1 patents, 1 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
US 6041334	A	20000321	US 1997960627	A	19971029	200023 B
			US 1999291263	A	19990413	

Priority Applications (no., kind, date): US 1997960627 A 19971029; US 1999291263 A 19990413

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6041334	A	EN	28	13	Division of application US 1997960627 Division of patent US 5983239

**Processing method for requests to access a data storage subsystem that maintains managed files in a storage management system**

**Alerting Abstract** ...NOVELTY - All managed files containing an identified user file are identified using a mapping table. The back - up of a first managed file is aborted if one or more managed files are identified in a target storage area . The first managed file is copied to the target storage area if there are no managed files identified in the target storage area . DESCRIPTION - A request for backing up a first managed file, stored in a data storage subsystem, to a target storage area is received . A user file , which occupies a predetermined position in the first managed file , is identified by utilizing a mapping table. INDEPENDENT CLAIMS are also included for the following...

...a signal-bearing medium; a data storage subsystem...

...USE - For processing requests to access a data storage subsystem that maintains managed files in a storage management system...

...ADVANTAGE - Enables conducting file management with reduced overhead by grouping smaller user files into larger aggregate files. Enables reclamation of wasted space between managed files and space which...

#### Class Codes

International Classification (Main): G06F-017/30

Manual Codes (EPI/S-X): T01-F05E ...

... T01-J05B1 ...

... T01-S03

## Original Publication Data by Authority

### Original Abstracts:

A **data storage** subsystem employs managed files comprising one or a contiguous aggregation of multiple constituent user **files**. A **mapping** table cross-references each managed **file** with the **names** and **locations** of its constituent user files. A storage table **cross - references** each managed file with its **address**. Eventually, "deleted-file space" arises as individual user files are deleted from managed files. "Reconstruction...

...space. Reconstruction preferably permits multiple embodiments of a managed file called "siblings". Reconstruction identifies contiguous **regions** of user files within a managed file, and copies these **regions** to adjacent **locations** in a target **area**. Before entering the reconstructed **file** in any tables, the **mapping** table is searched for a "paradigm" managed **file** containing the same user files as the reconstructed file. Finding a paradigm file, the storage table is modified by deleting reference to the pre-reconstruction filename, and adding an entry **cross - referencing** the paradigm file with the reconstructed file's storage **address**. Not finding the paradigm file, a new sibling filename is designated for the reconstructed **file**, an entry is added to the **mapping** table, **cross - referencing** the sibling with its constituent user files and their **locations** within the reconstructed file, reference to the pre-reconstruction file is deleted from the storage table, and an entry is added to the storage table, **cross - referencing** the sibling with its **address**. Finally, if the storage table has no other instances of the pre-reconstruction file, reference to the pre-reconstruction **file** is purged from the **mapping** table.

### Claims:

b Claim 5. /b A **data storage** subsystem, comprising: a storage including a **source** storage **area** and a target storage **area**, said storage containing one or more managed files each originally created with a contiguous aggregation of constituent user **files**; a **database**, including a **mapping** table cross-referencing each managed **file** with its constituent user files and a corresponding **location** of each user file within the managed file; a digital data processing apparatus coupled to the storage and the **database**; wherein the digital data processing apparatus is programmed to perform a method for processing requests to access **data** in the storage, the method comprising: **receiving** a request to **backup** a first managed **file** stored in the storage subsystem to a target storage **area**; utilizing the **mapping** table to identify a user **file** occupying a predetermined position in the first managed **file**; utilizing the **mapping** table to identify all managed **files** containing the identified user file; if one or more managed files are identified in the target storage **area**, aborting the **backup** of the first managed **file**; and if no managed files are identified in the target storage **area**, copying the first managed file to the target storage **area**.

57/3,K/77 (Item 71 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0012399142 - Drawing available  
WPI ACC NO: 2002-343038/200238  
XRPX Acc No: N2002-269789

**User data storage method in computer network, involves writing user data and file system type data in a file and transmitting file over communication link for remote data storage**

Patent Assignee: GOLD S (GOLD-I); HEWLETT-PACKARD CO (HEWP)

Inventor: GOLD S

**Patent Family** (3 patents, 2 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
GB 2365556	A	20020220	GB 200019015	A	20000804	200238 B
US 20020040405	A1	20020404	US 2001922082	A	20010803	200238 E
GB 2365556	B	20050427				200530 E

Priority Applications (no., kind, date): GB 200019015 A 20000804

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing	Notes
GB 2365556	A	EN	48	14		

200238

**User data storage method in computer network, involves writing user data and file system type data in a file and transmitting file over communication link for remote data storage**

#### Original Titles:

Gateway device for remote file server services...

...Gateway device for remote file server services

**Alerting Abstract** ...which emulates a file system type is created. The user data and file system type data are written in a data file and transmitted over a communication link, for remote data storage. ...Data generation method; Gateway appliance; Bulk data storage facility; Data storage provision method...

...ADVANTAGE - The problem of limited data capacity on communication links is solved satisfactorily and the cost of usage of bulk data repository facilities is reduced.

...DESCRIPTION OF DRAWINGS - The figure shows the relationship between bulk storage repository and single gateway.

**Title Terms...**/Index Terms/Additional Words: TRANSMIT ; ...

... LINK ;

#### Original Publication Data by Authority

#### Original Abstracts:

A bulk data repository b 201 /b for remote storage of bulk data

from a plurality of computer networks b 200 /b - b 207 /b is accessed over a plurality of communications links , e.g., the internet b 202 /b . Each computer network is provided with a gateway appliance b 200...

...emulates a file system, for example Windows NT(TM) or Novell NetWare(TM) by packaging data files to be stored in files for transmission over the communications linked to the data repository, each data file having appended a meta data header, which designates an address of the gateway appliance and a type of file system which the gateway appliance is emulating. The data repository, receives the data file with the meta data header, and stores the met data header locally in a local database prior to filing the data file . In a block of data reserved for the gateway appliance. The data repository can search data files by searching the meta data header to locate any of the data files of a gateway appliance. The data repository has automatic management tools for monitoring the amount of data storage space allocated to any gateway appliance, and for expanding the allocated data storage space if required.

**Claims:**

...entities, said method characterized by comprising the steps of:writing said user data to a local data storage area ( b 1001 /b ) in a said computer entity;creating an emulation data which emulates a file...

...type in use in said network;incorporating said user data and said file system type data in a data file for transmission ; andtransmitting said transmission file over a communications link for remote data storage.

57/3,K/146 (Item 140 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0008735029 - Drawing available  
WPI ACC NO: 1998-276970/  
XRPX Acc No: N1998-217879

**Information resource integration method for management of remote data base system connected to computer - involves generating conversion map between attributes stored in first and second database sample files based on which conversion process is performed in correlated manner during information inquiry**

Patent Assignee: AMERICAN TELEPHONE & TELEGRAPH CO (AMTT); AT & T CORP (AMTT); LUCENT TECHNOLOGIES INC (LUCE)

Inventor: ALLON I L; JOHAN J O; LEVY A Y; ORDILLE J J

**Patent Family** (4 patents, 3 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
JP 10091633	A	19980410	JP 1997189348	A	19970715	199825 B
CA 2208172	A	19980115	CA 2208172	A	19970618	199827 E
US 5778373	A	19980707	US 1996680090	A	19960715	199834 E
CA 2208172	C	20010116	CA 2208172	A	19970618	200107 E

Priority Applications (no., kind, date): US 1996680090 A 19960715

**Patent Details**

Number	Kind	Lan	Pg	Dwg	Filing	Notes
JP 10091633	A	JA	18	10		
CA 2208172	A	EN				
CA 2208172	C	EN				

**Information resource integration method for management of remote data base system connected to computer...**

...involves generating conversion map between attributes stored in first and second database sample files based on which conversion process is performed in correlated manner during information inquiry

**Original Titles:**

INTEGRATION OF INFORMATION RESOURCE OF INFORMATION SERVER

...

...Integration of an information server database schema by generating a translation map from exemplary files .

**Alerting Abstract** ...The method involves generating a first database sample file in which attributes are stored in a first database format. A second database sample file storing the attributes in a second database format is then generated. The attributes stored in the second file are generated corresponding to...

...A conversion map between the attributes stored in the first and second files, is generated. Corresponding to the...

...the conversion process is performed in correlated manner using the attributes based on the conversion map . Thus, the inquiry of information from the computer is accessed using the attributes stored in the remote database connected to the computer...



...ADVANTAGE - **Reduces** burden on **database** management system. Enables to inquire related information from **remote database** system.

**Title Terms...**/Index Terms/Additional Words: **CONNECT ; ...**

**...: MAP ; ...**

**... DATABASE ;**

#### **Original Publication Data by Authority**

#### **Original Abstracts:**

A method of **reducing** the burden on **database** administrators when integrating information from a **database** system with a computer system over a computer network is disclosed. Rather than requiring the **database** administrator to express the meaning of **database** attribute names in a new language, the **database** administrators needs only to specify **mappings** between **different database** schemas by creating **database** example files. The **database** example files contain a common body of information values stored using the corresponding attribute **names** of the **different database** schemas. The **database** example files then become the basis for generation of a translation **map** between the computer system and the **remote database** system. Then, information queries from a user are translated with the translation **map** to the **database** schema of the **remote database** system.

#### **Claims:**

A method of integrating information from a **database** system with a computer system, comprising the steps of: a) creating a **first database** example **file** comprising **first** attribute **names** having **first** values stored in a **first** format of a first **database** schema used in a **database** system; b) creating a **second database** example **file** comprising **second** attribute **names** having **second** values, stored in a **second** format of a second database schema, used in a computer system, said second values in...

...of said first database example file; c) generating a translation map between said first attribute **names** of said first **database** schema and said second attribute **names** of said second **database** schema, based on scoring possible **mappings** between said corresponding first and second values and formats; and d) performing translations using said **translation** map of information queries issued from said computer system and composed using **attribute** names of **said second** database schema, to information queries using **attribute** names of **said first** database schema, making **said** second values stored in **said** database system accessible to said computer system.

57/3,K/71 (Item 65 from file: 350)  
DIALOG(R) File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0012506330 - Drawing available  
WPI ACC NO: 2002-454283/200248  
XRPX Acc No: N2002-358357

**Automatic synchronization method for LOTUS NOTES database , involves generating mapping between document oriented and relational databases using portion of stored metadata**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); VELASCO G (VELA-I)  
Inventor: VELASCO G

**Patent Family** (2 patents, 1 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
US 20020059292	A1	20020516	US 199830257	A	19980225	200248 B
US 6446075	B1	20020903	US 199830257	A	19980225	200260 E

Priority Applications (no., kind, date): US 199830257 A 19980225

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20020059292	A1	EN	11	6	

**Automatic synchronization method for LOTUS NOTES database , involves generating mapping between document oriented and relational databases using portion of stored metadata**

#### Original Titles:

SYSTEM AND METHOD FOR AUTOMATICALLY SYNCHRONIZING **DIFFERENT** CLASSES OF **DATABASES** UTILIZING A **REPOSITORY DATABASE**

...

...System and method for automatically synchronizing **different** classes of **databases** utilizing a **repository database** .

**Alerting Abstract** ...NOVELTY - A **repository database** (102) is provided for storing **metadata** regarding a document oriented **database** and a relational **database** . The document oriented and relational **databases** are automatically generated using the respective portions of the stored **metadata** . A **mapping** between the document oriented and relational **databases** is automatically generated using a portion of the **metadata** .  
...Synchronization system; and Computer-readable medium storing **databases** synchronization program...

...USE - For automatically synchronizing **different** classes of **databases** such as DB/2 **database** , LOTUS NOTES **database** , etc...

...ADVANTAGE - Allows user to easily store and manipulate information in two **different** classes of **databases** , thereby **decreasing** development time and maintenance effort...

...DESCRIPTION OF DRAWINGS - The figure shows the block diagram of the **database** synchronization system...

...102 **Repository database**

**Title Terms**.../Index Terms/Additional Words: **DATABASE** ; ...

... MAP ;

#### Original Publication Data by Authority

##### Original Abstracts:

A system and method for synchronizing a **first database** and a **second database** is disclosed. The **first database** is of a **first class**, while the **second database** is of a **second class**. In one aspect, the method and system include providing a **repository database** for storing metadata regarding the **first database** and the **second database**, automatically generating the **first database** using a **first portion** of the **metadata** and automatically generating the **second database** using a **second portion** of the **metadata**. The method and system further include automatically generating a **mapping** between the **first database** and the **second database** using a **third portion** of the **metadata** stored in the **repository database**. In a **second aspect**, the **first database** is a **preexisting database**. In this aspect, the method and system include providing a **repository database** for storing metadata regarding the **first database** and the **second database** and automatically generating the **second database** using a **first portion** of the **metadata**. In this aspect, the method and system further include automatically generating a **mapping** between the **first database** and the **second database** using a **second portion** of the **metadata** stored in the **repository database**.

...

...A system and method for synchronizing a **first database** and a **second database** is disclosed. The **first database** is of a **first class**, while the **second database** is of a **second class**. The method and system include providing a **repository database** for storing metadata regarding the **first database** and the **second database**, automatically generating the **first database** using a **first portion** of the **metadata** and automatically generating the **second database** using a **second portion** of the **metadata**. The method and system further include automatically generating a **mapping** between the **first database** and the **second database** using a **third portion** of the **metadata** stored in the **repository database**.

##### Claims:

What is claimed is: b 1 /b . A method for synchronizing a **first database** of a **first class** and a **second database** of a **second class** comprising the steps of: (a) providing a **repository database** for storing metadata regarding the **first database** and the **second database**; (b) automatically generating the **first database** using a **first portion** of the **metadata**; (c) automatically generating the **second database** using a **second portion** of the **metadata**; and (d) automatically generating a **mapping** between the **first database** and the **second database** using a **third portion** of the **metadata** stored in the **repository database**.

...

...A method for synchronizing a **first database** of a **first class** and a **second database** of a **second class** comprising the steps of: (a) providing a **repository database** for storing metadata regarding the **first database** and the **second database** wherein the **first database** is of a **different type** than the **second database** and **metadata** comprises

information describing the structure of a database ;(b) automatically generating the first database using a first portion of the metadata ;(c) automatically generating the second database using a second portion of the metadata ; and(d) automatically generating a mapping between the first database and the second database using a third portion of the metadata stored in the repository database.

57/3,K/48 (Item 42 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013266524 - Drawing available  
WPI ACC NO: 2003-352309/  
XRPX Acc No: N2003-281361

Metadata stream generation method in data processing system, involves placing attribute call associated with primitive call, and primitive call, in metadata stream

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: COHEN M L; COOPER M R; NOGAY P E; VANDERWIELE M W

Patent Family (1 patents, 1 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
US 6510426	B1	20030121	US 1998163915	A	19980930	200333 B

Priority Applications (no., kind, date): US 1998163915 A 19980930

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6510426	B1	EN	14	10	

Metadata stream generation method in data processing system, involves placing attribute call associated with primitive call, and primitive call, in metadata stream

#### Original Titles:

Method and apparatus for compacting a metadatas stream in a data processing system

**Alerting Abstract** ...NOVELTY - An attribute call is received to set an attribute. In response, a first database is queried to determine whether the attribute call is present in the first database. The first database is updated in response to the attribute call being absent in the first database. In response to receiving a primitive call, an attribute call is associated with the primitive is received from the first database. ...made as to whether the attribute call associated with the primitive is present in a second database. The second database is updated to include the attribute call associated with the primitive call in response to the attribute call associated with the primitive call being absent in the second database. The attribute call associated with the primitive call, and the primitive call, are placed in the metadata stream...

...USE - For generating metadata stream in data processing system used in many types of applications, such as spreadsheets, graphical applications, image applications, and front ends to databases.

...

...ADVANTAGE - Attribute calls can be effectively managed in either the creation of metadata stream or via a post processor to significantly reduce the overhead required to process the metadata stream

#### Class Codes

International Classification (Main): G06F-017/00

Manual Codes (EPI/S-X): T01-J05B1 ...



... T01-J05B4M ...

... T01-S03

#### Original Publication Data by Authority

#### Original Abstracts:

A method and apparatus in a **data** processing system for generating a metadata stream. An attribute call is **received** to set an attribute. In response, a **first database** is queried to determine whether the attribute call is present in the **first database**. The **first database** is **updated** in response to the attribute call being absent in the **first database**. In response to **receiving** a primitive call, an attribute call associated with the primitive is retrieved from the **first database**. A determination is made as to whether the attribute call associated with the primitive is present in a **second database**. The **second database** is; **updated** to include the attribute call associated with the primitive call in response to the attribute call associated with the primitive call being absent in the **second database**. The attribute call associated with the primitive call and the primitive call is placed in the **metadata** stream.

#### Claims:

What is claimed is: 1. A method in a **data** processing system for generating a **metadata** stream, the method comprising the **data** processing system implemented steps of: **receiving** an attribute call to set an attribute; querying a first group of attributes to determine whether the attribute call is present and identical within the first group of attributes; **updating** the group of attributes in response to an absence of a determination that the attribute call is present and identical to an attribute call in the first group attributes; **receiving** a primitive call; retrieving an attribute associated with the primitive from the first group of...

...primitive is present and identical to an attribute call in a second group of attributes; **updating** the second group of attributes to include the attribute call associated with the primitive call...

...and placing the attribute call associated with the primitive call and the primitive call in **the** metadata stream.

57/3,K/47 (Item 41 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2006 The Thomson Corporation. All rts. reserv.

0013267078 - Drawing available

WPI ACC NO: 2003-352879/200333

Related WPI Acc No: 2005-657045

XRPX Acc No: N2003-281826

**Data management system has image conversion processor which converts data files that are organized into data slices including ID number and descriptor, into image files**

Patent Assignee: KROLL ONTRACK INC (KROL-N); ONTRACK DATA INT INC (ONTR-N)

Inventor: BLACK C; BROCKWAY S M; CRAIG R M; PARTINGTON T; SCHMIDT R A

**Patent Family** (5 patents, 99 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20030004922	A1	20030102	US 2001894373	A	20010627	200333 B
WO 2003003253	A2	20030109	WO 2002US17895	A	20020606	200333 E
EP 1428145	A2	20040616	EP 2002741871	A	20020606	200439 E
			WO 2002US17895	A	20020606	
AU 2002314942	A1	20030303	AU 2002314942	A	20020606	200452 E
AU 2002314942	A8	20051027	AU 2002314942	A	20020606	200624 E

Priority Applications (no., kind, date): US 2001894373 A 20010627

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20030004922	A1	EN	15	8	
WO 2003003253	A2	EN			

National Designated States,Original: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SD SE SG SI SK SL TJ TM TN TR TT TZ UA UG UZ VN YU ZA ZM ZW

Regional Designated States,Original: AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZM ZW

EP 1428145 A2 EN PCT Application WO 2002US17895  
Based on OPI patent WO 2003003253

Regional Designated States,Original: AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI TR

AU 2002314942 A1 EN Based on OPI patent WO 2003003253  
AU 2002314942 A8 EN Based on OPI patent WO 2003003253

**Data management system has image conversion processor which converts data files that are organized into data slices including ID number and descriptor, into image files**

**Alerting Abstract ...NOVELTY** - A file organizing processor (24) organizes data files received from an input processor (22), into data slices including ID number and descriptor. A logging processor (28) logs the data files in a local database (26) that is uploaded to a global database (30). An image conversion processor (36) converts the data files into image files that is exported by an output processor (38)....USE - For managing data file of different format such as Microsoft Mail , Outlook, GroupWise, Lotus Notes, Word, Excel, PowerPoint and Access stored in hard drive, floppy disk, backup tape, CD, optical

device, etc., operated by different operating system such as UNIX, NOVELL, NT...

...ADVANTAGE - Automated data management system is provided for logging, processing and reporting large volume of **data files** in an efficient manner. The time required to generate a report containing organized image **files**, is substantially **reduced**, hence the quality and efficiency of the **data files** are improved. Duplication of **data files** is eliminated and size of the **data files** are **reduced**, thus parallel processing of the **data files** is enabled...

...26 local database

...

...30 global database

Title Terms.../Index Terms/Additional Words: ID ;

Original Publication Data by Authority

#### Original Abstracts:

...different versions, stored on different media, and/or run by different operating systems, includes a **first** processor for restoring a plurality of **received data files**, the **data files** being capable of being **different** file types; a **file** organizing/categorizing processor for organizing the **received data files** into **data slices**, each **data slice** including an identification number and a descriptor that describes characteristics of the **received data file**; a **file** logging processor for logging the **received data files** into a **first database** based on the **data slices**; a **data uploading** processor for **uploading** the **first database** to a **second database**; a de-duplicate processor for calculating a SHA value of the **received data files** to determine whether the **received data files** have duplicates and flagging duplicated **data files** in the **second database**; an image conversion processor for converting at least a portion of the **received data files** into **image files**; and a second processor for **exporting** the **image files**.

...

...different versions, stored on different media, and/or run by different operating systems, includes a **first** processor for restoring a plurality of **received data files**, the **data files** being capable of being **different** file types; a **file** organizing/categorizing processor for organizing the **received data files** into **data slices**, each **data slice** including an identification number and a descriptor that describes characteristics of the **received data file**; a **file** logging processor for logging the **received data files** into a **first database** based on the **data slices**; a **data uploading** processor for **uploading** the **first database** to a **second database**; a de-duplicate processor for calculating a SHA value of the **received data files** to determine whether the **received data files** have duplicates and flagging duplicated **data files** in the **second database**; an image conversion processor for converting at least a portion of the **received data files** into **image files**; and a second processor for **exporting** the **image**

files .

...

...different versions, stored on different media, and/or run by different operating systems, includes a **first** processor for restoring a plurality of **received data files**, the **data files** being capable of being **different** file types; a **file** organizing/categorizing processor for organizing the **received data files** into **data** slices, each **data** slice including an identification number and a descriptor that describes characteristics of the **received data file**; a **file** logging processor for logging the **received data files** into a first **database** based on the **data** slices; a **data** uploading processor for uploading the **first database** to a **second database**; a de-duplicate processor for calculating a SHA value of the **received data files** to determine whether the **received data files** have duplicates and flagging duplicated **data files** in the **second database**; an image conversion processor for converting at least a portion of the **received data files** into **image files**; and a second processor for **exporting** the **image files**.

...

...des fichiers de donnees recus en fichiers d'images; et un second processeur permettant d'**exporter** les fichiers d'images.

**Claims:**

What is claimed is: b 1 /b . A data management system, comprising: a **first** processor for restoring a plurality of **received data files**, the **data files** being capable of being **different** file types; a **file** organizing/categorizing processor, coupled to the **first** processor, for organizing the **received data files** into **data** slices, each **data** slice including an identification number and a descriptor that describes characteristics of the **received data file**; a **file** logging processor, coupled to the **file** organizing/categorizing processor, for logging the received data files into a first database based on...

...processor, coupled to the data uploading processor, for calculating a SHA value of the received **data files** to determine whether the **received data files** have duplicates and flagging duplicated **data files** in the **second database**; an image conversion processor, coupled to the de-duplicate processor, for converting at least a portion of the **received data files** into image files; and a second processor, coupled to the image conversion processor, for **exporting** the **image files**.

57/3,K/41 (Item 35 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

0013351636 - Drawing available  
WPI ACC NO: 2003-439497/200341  
Related WPI Acc No: 1998-297283; 1999-166153; 2000-023690; 2005-434102  
XRPX Acc No: N2003-350693

Hyperlink reference tracking/re-directing method in client/ server  
computer system, involves selecting information including tracking resource  
locator and data, to make client use tracking and content resource  
locators

Patent Assignee: INFOSEEK CORP (INFO-N)

Inventor: KIRSCH S T; LINDBLAD C J

Patent Family (2 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20030046361	A1	20030306	US 1996604468	A	19960221	200341 B
			US 199871674	A	19980501	
			US 2000655999	A	20000606	
			US 2002190341	A	20020703	
US 6859833	B2	20050222	US 2002190341	A	20020703	200515 E

Priority Applications (no., kind, date): US 2000655999 A 20000606; US  
199871674 A 19980501; US 1996604468 A 19960221; US 2002190341 A  
20020703

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20030046361	A1	EN	14	7	Continuation of application US
1996604468					Continuation of application US
199871674					Continuation of application US
2000655999					Continuation of patent US 5751956
					Continuation of patent US 6189030
					Continuation of patent US 6466966

Hyperlink reference tracking/re-directing method in client/ server  
computer system, involves selecting information including tracking resource  
locator and data, to make client use tracking and content resource  
locators

#### Original Titles:

Method and apparatus for redirection of server external hyper- link  
references...

...Method and apparatus for redirection of server external hyper- link  
references

**Alerting Abstract** ...The informational element is identified on the  
client system, with an information obtained from a content server  
through a content resource locator. The selection of the informational  
element including tracking resource locator and accounting data, causes  
the client to use the tracking and content resource locators to provide  
data to the tracking server, and to obtain data from content server  
respectively....USE - For tracking and re-directing hyperlink reference



in client and **server** computer systems **connected** through Internet...

...ADVANTAGE - The reference **identifier** and a **redirection** directive are maintained with the URL specification provided by the client. Thus increases the efficiency of tracking method. Eliminates the need of multiple external **data** references. Thus **minimizes** the CPU and disk intensive load on the web **server** computer system...

...DESCRIPTION OF DRAWINGS - The figure shows a flow diagram explaining the **hyperlink** reference tracking/re-directing method.

**Title Terms...**/Index Terms/Additional Words: **CONTENT**

#### **Class Codes**

International Classification (Main): **G06F-015/16 ...**

**... G06F-015/173**

Manual Codes (EPI/S-X): **T01-N01A2C ...**

**... T01-N02B1A ...**

**... T01-N02B2A**

#### **Original Publication Data by Authority**

#### **Original Abstracts:**

A message is provided to a tracking **server** system in response to a client system referencing a predetermined resource locator that corresponds to a resource external to the tracking **server** system. The tracking **server** system indirectly provides for the client system to have an informational element selectable by the client system, where the informational element is graphically identified on the client system with informational **content** obtainable from a **content server** system through use of a **content** resource locator. The informational element includes a tracking resource locator, referencing the tracking **server** system, and data identifying the informational element. The selection of the informational element causes the client system to use the tracking resource locator to provide the data to the tracking **server** system and to use the **content** resource locator to obtain the informational **content** from the **content server** system...

...A message is provided to a tracking **server** system in response to a client system referencing a predetermined resource locator that corresponds to a resource external to the tracking **server** system. The tracking **server** system indirectly provides for the client system to have an informational element selectable by the client system, where the informational element is graphically identified on the client system with informational **content** obtainable from a **content server** system through use of a **content** resource locator. The informational element includes a tracking resource locator, referencing the tracking **server** system, and data identifying the informational element. The selection of the informational element causes the client system to use the tracking resource locator to provide the data to the tracking **server** system and to use the **content** resource locator to obtain the informational **content** from the **content server** system.

#### **Claims:**

b 1 /b . A method of providing a message to a tracking **server** system in

response to a client system referencing a predetermined resource locator that corresponds to a resource external to said **server** system, said method comprising the steps of: a) providing for a client system to have...

...client system, wherein said informational element is graphically identified on said client system with informational **content** obtainable from a first server system through use of a first resource locator, and wherein said informational element includes a **second** resource locator referencing a **second server** system and data identifying said informational element; b) providing for said client system to use said second resource locator to provide said data to said **second server** system in response to the selection of said informational element; and c) providing for said client system to use said first resource locator to obtain said **informational** content from said **first** server system in response to the selection of said informational element...

...What is claimed is: 1. A method of providing a message to a **tracking** server system in response to a client system referencing a predetermined resource locator that corresponds to a resource external to said server system, said method comprising the steps of: a) providing for a client system to...

...said client system, wherein said informational element is graphically identified on said client system with **informational** content obtainable from a **first** server system through use of a first resource locator, and wherein said informational element includes a second resource locator referencing a **second** server system and data identifying said informational element; b) providing for said client system with said second resource locator to provide said data to said **second** server system in response to the selection of said informational element; and c) providing for said client system to use said first resource locator to obtain said informational content **from** said first server system in response to the selection of said informational element.

57/3,K/21 (Item 15 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2006 The Thomson Corporation. All rts. reserv.

*THIS  
APPLICATION*

0014110824 - Drawing available  
WPI ACC NO: 2004-295139/200427  
Related WPI Acc No: 2004-270155; 2004-270171; 2004-270184  
XRPX Acc No: N2004-234412

**Files set transferring method for data storage system e.g. magnetic disk drive, involves replacing stub file for specified file 's full content if full content of specified file is not transferred , upon receipt of client request**

Patent Assignee: EVILIA S H (EVIL-I); EXAGRID SYSTEMS INC (EXAG-N);  
GILHOOLY S R (GILH-I); HANSEN T G (HANS-I); KENNA H R (KENN-I);  
POWELL J E (POWN-I); THERRIEN D G (THER-I); VANDERSPEK A (VAND-I)  
Inventor: EVILIA S H; GILHOOLY S R; HANSEN T G; KENNA H R; POWELL J E;  
THERRIEN D G; VANDERSPEK A

**Patent Family** (5 patents, 105 countries)

Patent			Application			
Number	Kind	Date	Number	Kind	Date	Update
WO 2004025404	A2	20040325	WO 2003US28250	A	20030910	200427 B
US 20040088382	A1	20040506	US 2002409684	P	20020910	200430 E
			US 2003659642	A	20030910	
AU 2003268572	A1	20040430	AU 2003268572	A	20030910	200462 E
EP 1540441	A2	20050615	EP 2003749544	A	20030910	200539 E
			WO 2003US28250	A	20030910	
JP 2005538469	W	20051215	WO 2003US28250	A	20030910	200582 E
			JP 2004536389	A	20030910	

Priority Applications (no., kind, date): US 2003659642 A 20030910; US  
2002409684 P 20020910

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 2004025404	A2	EN	32	13	
National Designated States,Original: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW					
Regional Designated States,Original: AT BE BG CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW					
US 20040088382	A1	EN			Related to Provisional US 2002409684
AU 2003268572	A1	EN			Based on OPI patent WO 2004025404
EP 1540441	A2	EN			PCT Application WO 2003US28250 Based on OPI patent WO 2004025404
Regional Designated States,Original: AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LT LU LV MC MK NL PT RO SE SI SK TR					
JP 2005538469	W	JA	20		PCT Application WO 2003US28250 Based on OPI patent WO 2004025404

**Files set transferring method for data storage system e.g. magnetic disk drive, involves replacing stub file for specified file 's full content if full content of specified file is not transferred , upon receipt of client request**

**Original Titles:**

METHOD AND APPARATUS FOR **SERVER SHARE MIGRATION** AND **SERVER RECOVERY** USING **HIERARCHICAL STORAGE MANAGEMENT...**

...VERFAHREN UND VORRICHTUNG ZUR **SERVER -SHARE- MIGRATION** UND **SERVER -BEHEBUNG** UNTER VERWENDUNG **HIERARCHISCHER SPEICHERUNGSVERWALTUNG...**

...METHOD AND APPARATUS FOR **SERVER SHARE MIGRATION** AND **SERVER RECOVERY** USING **HIERARCHICAL STORAGE MANAGEMENT...**

...PROCEDE ET DISPOSITIF DE **MIGRATION** DE RESSOURCES PARTAGEES DE SERVEUR AU MOYEN D'UNE GESTION MEMOIRE HIERARCHIQUE...

...Method and apparatus for **server share migration** and **server recovery** using **hierarchical storage management...**

...METHOD AND APPARATUS FOR **SERVER SHARE MIGRATION** AND **SERVER RECOVERY** USING **HIERARCHICAL STORAGE MANAGEMENT...**

...PROCEDE ET DISPOSITIF DE **MIGRATION** DE RESSOURCES PARTAGEES DE SERVEUR AU MOYEN D'UNE GESTION MEMOIRE HIERARCHIQUE

**Alerting Abstract** ...NOVELTY - The method involves **updating** a location component in a **destination file server** . A **stub file** is replaced for a specified file's full **content** if full **content** of the specified **file** is not **transferred** , upon receipt of a client request for the specified **file** . The **stub file** for the specified **file** is a higher priority task than replacing the **stub files** for non-requested **files** .  
...USE - Used for computer **primary data storage** system e.g. magnetic disk drive, magnetic tape drive and optical disk drive...

...ADVANTAGE - The method provides effective **data** protection while **reducing** the time necessary for clients to regain access to their data after complete loss of a **file server** .

...  
...DESCRIPTION OF DRAWINGS - The drawing shows a **data storage** system across three data centers...

...3 **Repositories**

...

...4 **File server**

**Title Terms**.../Index Terms/Additional Words: **TRANSFER ; ...**

... **STUB ; ...**

... **CONTENT ;**

**Class Codes**

International Classification (Main): **G06F , ...**

... **G06F-001/00 ...**

... **G06F-012/00 ...**

... **G06F-015/16**

Manual Codes (EPI/S-X): **T01-C01A ...**

... T01-G03 ...

... T01-H01B1 ...

... T01-H01B2 ...

... T01-N02B1A

#### Original Publication Data by Authority

#### Original Abstracts:

The present invention relates to computer **primary data storage** systems and methods that provide comprehensive data protection...

...The present invention relates to computer **primary data storage** systems and methods that provide comprehensive data protection...

...The present invention relates to computer **primary data storage** systems and methods that provide comprehensive data protection...

#### Claims:

What is claimed is: b 1 /b . A method for **transferring** a set of **files** , the method comprising: **receiving metadata** and **stub files** associated with the set of **files** at a destination **fileserver** ; **updating** a location component in the **destination fileserver** to maintain a list of **repository** nodes that are associated with each file in the set of **files** ; replacing each **stub file** with the full **content** of the **file** associated with the **stub file** ; and while replacing **each stub file** , upon receipt of a client request for a specified file in the set of files, if the **full content** of the **specified file** has not yet **been** transferred, then replacing **the stub file** for the **specified file** with the specified file's **full content** , wherein replacing **the stub file** for the **specified file** is a higher priority task than replacing **the stub files** for non- **requested files** .